

The Parameterized Complexity of the Minimum Shared Edges Problem*

Till Fluschnik¹, Stefan Kratsch², Rolf Niedermeier¹, and Manuel Sorge¹

¹Institut für Softwaretechnik und Theoretische Informatik, TU Berlin, Germany, {till.fluschnik, rolf.niedermeier, manuel.sorge}@tu-berlin.de

²Institut für Informatik, Universität Bonn, Germany, kratsch@cs.uni-bonn.de

February 5, 2016

Abstract

We study the NP-complete MINIMUM SHARED EDGES (MSE) problem. Given an undirected graph, a source and a sink vertex, and two integers p and k , the question is whether there are p paths in the graph connecting the source with the sink and sharing at most k edges. Herein, an edge is shared if it appears in at least two paths. We show that MSE is W[1]-hard when parameterized by the treewidth of the input graph and the number k of shared edges combined. We show that MSE is fixed-parameter tractable with respect to p , but does not admit a polynomial-size kernel (unless $\text{NP} \subseteq \text{coNP/poly}$). In the proof of the fixed-parameter tractability of MSE parameterized by p , we employ the treewidth reduction technique due to Marx, O’Sullivan, and Razgon [ACM TALG 2013].

Key words: Fixed-parameter tractability; W-hardness; kernelization; tree decompositions of graphs; treewidth reduction technique; VIP routing.

1 Introduction

We consider the parameterized complexity of the following basic routing problem.

MINIMUM SHARED EDGES (MSE)

Input: A graph $G = (V, E)$, $s, t \in V$, $p \in \mathbb{N}$ and $k \in \mathbb{N}_0$.

Question: Is there a (p, s, t) -routing in G in which at most k edges are shared?

Herein, a (p, s, t) -routing is a set of s - t paths with cardinality p , and an edge is called *shared* if it is contained in at least two of the paths in the routing. If s and t are understood from the context, we simplify notation and speak of a p -routing and call the paths it contains *routes*. MINIMUM SHARED EDGES is polynomial-time solvable with $k = 0$, while it becomes NP-hard for general values of k [14].

MINIMUM SHARED EDGES has two natural applications. One is to route an important person which is under threat of attack from s to t in a street network. In order to confound attackers, $p - 1$ additional, empty convoys are routed, and guards are placed on streets that are shared by routes. MINIMUM SHARED EDGES then minimizes the costs to place guards [24].

A second application arises from finding a resilient way of communication between two servers s and t in an interconnection network, assuming that $p - 1$ faulty connections may be present that block or alter the communicated information. Finding p edge-disjoint paths ensures at least one piece of information arrives unscathed. When this is not possible, and if we can ensure that a link

*Till Fluschnik, Stefan Kratsch, and Manuel Sorge gratefully acknowledge support by the DFG, projects DAMM (NI 369/13-2), PRE- MOD (KR 4286/2-1), and DAPA (NI 369/12-2), respectively. An extended abstract appeared in: Proceedings of the 35th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2015), volume 45, pages 448–462.

is not faulty by expending some fixed cost per link, then MINIMUM SHARED EDGES is the problem of finding a resilient way of communication that minimizes the overall costs [25].

We study MINIMUM SHARED EDGES from a parameterized complexity perspective, that is, for certain parameters ℓ of the inputs (of size r), we identify algorithms with running time $f(\ell) \cdot \text{poly}(r)$ or we prove that such algorithms are unlikely to exist. There are two natural parameters for MINIMUM SHARED EDGES: the number p of routes and the number k of shared edges. Both of them can be reasonably assumed to be small in applications. As we will see, there is also a connection between p and the treewidth tw of G .

Related Work. Omran et al. [24] introduced MINIMUM SHARED EDGES on directed graphs and showed NP-hardness by a reduction from SET COVER. The reduction also implies W[2]-hardness with respect to the number k of shared arcs in this directed case. Undirected MINIMUM SHARED EDGES admits an XP-algorithm with respect to treewidth, more specifically, it can be solved in $O((n + m) \cdot (p + 1)^{2^{\omega \cdot (\omega + 1)/2}})$ time [28].

Assadi et al. [2] introduced a generalization of directed MINIMUM SHARED EDGES, called MINIMUM VULNERABILITY, which additionally considers arc weights (the cost of sharing an arc), arc capacities (an upper bound on the number of routes supported by an arc) and a share-threshold for each arc (the threshold of routes, possibly other than two, after which the arc becomes shared). Directed MINIMUM VULNERABILITY admits an XP-algorithm with respect to the number p of routes [2]. Undirected MINIMUM VULNERABILITY is NP-hard even on bipartite series-parallel graphs, but admits a pseudo-polynomial-time algorithm on bounded treewidth graphs [1]. Furthermore, MINIMUM VULNERABILITY is fixed-parameter tractable with respect to p on chordal graphs [1]. Recently, Fluschnik and Sorge [15] showed that MSE remains NP-hard on planar graphs of maximum degree four.

There are also several results regarding approximation algorithms and lower bounds [2, 24]; however, our focus is on exact algorithms.

Our Contributions. First we show that MINIMUM SHARED EDGES is NP-complete and W[2]-hard with respect to the number k of shared edges (Section 3). We then prove two main results, namely, that MINIMUM SHARED EDGES is fixed-parameter tractable (FPT) with respect to the number p of routes (Section 5) and that it is W[1]-hard with respect to the treewidth tw and the number k of shared edges combined (Section 7). Moreover, complementing the fixed-parameter tractability result with respect to p , we show that there is no polynomial-size problem kernel with respect to p (Section 6).

The FPT result with respect to p is obtained by modifying the input graph so that the resulting graph has treewidth bounded by some (exponential) function of p using the treewidth reduction technique [21] (see Section 5). Then we apply a dynamic program which also is an FPT algorithm with respect to p and tw (Section 4). For this purpose, we design a new dynamic program rather than using the ones from the literature [1, 28]. In comparison, ours yields an improved running time in the FPT algorithm with respect to p , that is, the dependence is doubly exponential on p rather than triply exponential. Our result complements the known FPT algorithm for undirected MINIMUM VULNERABILITY on chordal graphs, parameterized by p . Treewidth reduction has lately also found application in a wide variety of problems, for example, in graph coloring [5], graph partitioning [3], and arc routing [17].

As mentioned, our second main result is that MINIMUM SHARED EDGES is W[1]-hard with respect to the treewidth tw and the number k of shared edges combined. This provides a corresponding lower bound for the known polynomial-time algorithms on constant treewidth graphs for MINIMUM SHARED EDGES and for the more general undirected MINIMUM VULNERABILITY [1, 28]. More precisely, the exponents in the running time depend on tw and our result shows that removing this dependence is impossible unless $\text{FPT} = \text{W}[1]$. Interestingly, the known dynamic programs on tree decompositions keep track of the number of routes over certain separators in their tables. Our hardness result shows that information of this sort is crucial, that is, it is unlikely that there are dynamic programs with table entries relying only on information bounded by the treewidth.

2 Preliminaries

We use standard notation from parameterized complexity [6, 10, 13, 22] and graph theory [8, 27].

Graphs and Tree Decompositions. Unless stated otherwise, all graphs are without parallel edges or loops. When it is not ambiguous, we use n for the number of vertices of a graph and m for the number of edges.

Let $G = (V, E)$ be an undirected graph. We write $V(G)$ for the vertex set of graph G and $E(G)$ for the edge set of graph G . We define the size of graph G as $|G| := |V(G)| + |E(G)|$. For a vertex set $W \subseteq V(G)$, we denote by $G[W]$ the subgraph of G with vertex set $\{v \in V(G) \mid v \in W\}$ and edge set $\{\{v, w\} \in E(G) \mid v, w \in W\}$. We say that $G[W]$ is the subgraph of G *induced* by the vertex set W . For an edge set $F \subseteq E(G)$, we denote by $G[F]$ the subgraph of G with vertex set $\{v \in V(G) \mid (e \in F) \wedge (v \in e)\}$ and edge set $\{e \in E(G) \mid e \in F\}$. We say that $G[F]$ is the subgraph of G *induced* by the edge set F . For an edge $e \in E$, we denote by $G/\{e\}$ the contraction of edge e in G , and we denote by $G \setminus \{e\}$ the deletion of edge e in G (we write G/e and $G \setminus e$ for short). Consequently, for a set of edges $F \subseteq E$ we write G/F and $G \setminus F$ for the contraction and the deletion of the edges in F , respectively. We write $\Delta(G)$ to denote the maximum degree of graph G and $\text{diam}(G)$ to denote the diameter of G .

A *tree decomposition* of a graph G is a tuple $\mathbb{T} := (T, (B_\alpha)_{\alpha \in V(T)})$ of a tree T and family $(B_\alpha)_{\alpha \in V(T)}$ of sets $B_\alpha \subseteq V(G)$, called *bags*, such that $V(G) = \bigcup_{\alpha \in V(T)} B_\alpha$,

- (i) for every edge $e \in E(G)$ there exists an $\alpha \in V(T)$ such that $e \subseteq B_\alpha$ and
- (ii) for each $v \in V(G)$, the graph induced by the node set $\{\alpha \in V(T) \mid v \in B_\alpha\}$ is a tree.

The *width* ω of a tree decomposition \mathbb{T} of a graph G is defined as $\omega(\mathbb{T}) := \max\{|B_\alpha| - 1 \mid \alpha \in V(T)\}$. The *treewidth* $\text{tw}(G)$ of a graph G is the minimum width over all tree decompositions of G . A tree decomposition $\mathbb{T} = (T, (B_\alpha)_{\alpha \in V(T)})$ is a *nice tree decomposition with introduce edge nodes* if the following conditions hold.

- (i) The tree T is rooted and binary.
- (ii) For all edges in $E(G)$ there is exactly one *introduce edge node* in \mathbb{T} , where an introduce edge node is a node α in the tree decomposition \mathbb{T} of G labeled with an edge $\{v, w\} \in E(G)$ with $v, w \in B_\alpha$ that has exactly one child node α' ; furthermore $B_\alpha = B_{\alpha'}$.
- (iii) Each node $\alpha \in V(T)$ is of one of the following types:
 - *introduce edge node*;
 - *leaf node*: α is a leaf of T and $B_\alpha = \emptyset$;
 - *introduce vertex node*: α is an inner node of T with exactly one child node $\beta \in V(T)$; furthermore $B_\beta \subseteq B_\alpha$ and $|B_\alpha \setminus B_\beta| = 1$;
 - *forget node*: α is an inner node of T with exactly one child node $\beta \in V(T)$; furthermore $B_\alpha \subseteq B_\beta$ and $|B_\beta \setminus B_\alpha| = 1$;
 - *join node*: α is an inner node of T with exactly two child nodes $\beta, \gamma \in V(T)$; furthermore $B_\alpha = B_\beta = B_\gamma$.

A given tree decomposition can be modified in linear time to fulfill the above constraints; moreover, the number of nodes in such a tree decomposition of width ω is $O(\omega \cdot n)$ [7, 20].

Flows, Cuts, and Paths. Let G be an undirected, connected graph. A *cut* $C \subseteq E$ is a set of edges such that the graph $G \setminus C$ is not connected. Let $s, t \in V(G)$ be two vertices in G . An *s-t cut* C is a cut such that the vertices s and t are not connected in $G \setminus C$. A *minimum s-t cut* is an *s-t cut* C such that $|C| = \min |C'|$, where the minimum is taken over all *s-t cuts* C' in G . An *s-t cut* C in G is *minimal* if for all edges $e \in C$ it holds that $C \setminus \{e\}$ is not an *s-t cut* in G .

A *path* is a connected graph with exactly two vertices of degree one and no vertex of degree at least three. We call the vertices with degree one the *endpoints* of the path. The *length* of a path is defined as the number of edges in the path. For two distinct vertices $s, t \in V(G)$, we refer to the path with endpoints s and t (as subgraph of G) as *s-t path* in G . An *s-t path* in G is a *shortest s-t path* in G if there is no *s-t path* in G of smaller length. We denote by $\text{dist}_G(s, t)$ the length of a shortest *s-t path* in G .

A graph G has *edge capacities* if there is a function $c : E(G) \rightarrow \mathbb{R}_{\geq 0}$ that maps each edge in G to a number in $\mathbb{R}_{\geq 0}$, where $\mathbb{R}_{\geq 0}$ denotes the non-negative real numbers. For an edge $e \in E(G)$, we say that $c(e)$ is the *capacity* of edge e in G . We say that graph G has *unit edge capacities* if

$c(e) = 1$ for all $e \in E(G)$. In this work, if we consider a graph with edge capacities, then we always consider a graph with unit edge capacities.

Let D be an directed graph with edge capacities $c : E(D) \rightarrow \mathbb{R}_{\geq 0}$ and let $s, t \in V(D)$ be two vertices in D . An s - t flow in D is a function $f : E(D) \rightarrow \{0, 1\}$ such that

- (i) $f(e) \leq c(e)$ for all $e \in E(D)$,
- (ii) $\sum_{w \in V(D): (v, w) \in E(D)} f((v, w)) = \sum_{w \in V(D): (w, v) \in E(D)} f((w, v))$ for all $v \in V(D) \setminus \{s, t\}$, and
- (iii) $\sum_{w \in V(D): (w, t) \in E(D)} f((w, t)) - \sum_{w \in V(D): (t, w) \in E(D)} f((t, w)) \geq 0$.

The *value* of an s - t flow f in D is defined as

$$|f| := \sum_{w \in V(D): (w, t) \in E(D)} f((w, t)) - \sum_{w \in V(D): (t, w) \in E(D)} f((t, w)).$$

An s - t flow f is a *maximum s - t flow* in D if there is no s - t flow f' in D with $|f'| > |f|$.

For an undirected graph G we call the directed graph D_G the directed version of graph G if $V(D_G) = V(G)$ and $E(D_G) = \{(u, v), (v, u) \mid \{u, v\} \in E(G)\}$. If G has edge capacities $c : E(G) \rightarrow \mathbb{R}_{\geq 0}$, then D_G has edge capacities $c' : E(D_G) \rightarrow \mathbb{R}_{\geq 0}$ with $c'((u, v)) := c'((v, u)) := c(\{u, v\})$ for all edges $\{u, v\} \in E(G)$. We say that a function $f : E(G) \rightarrow \{0, 1\}$ is an s - t flow with value $|f| := \sum_{w \in V(G): \{w, t\} \in E(G)} f(\{w, t\})$ in an undirected graph G with edge capacities $c : E(G) \rightarrow \mathbb{R}_{\geq 0}$ and $s, t \in V(G)$, if there is an s - t flow f' in D_G such that $|f'| = |f|$, and for all edges $\{u, v\} \in E(G)$ it holds that $f'((u, v)) = 0$ and $f'((v, u)) = f(\{u, v\})$, or $f'((v, u)) = 0$ and $f'((u, v)) = f(\{u, v\})$. An s - t flow f_1 is a *maximum s - t flow* in G if there is no s - t flow f_2 in G with $|f_2| > |f_1|$. We remark that our definition of s - t flows is close to the definition given by Goldberg and Rao [16]. For more information on flows, in particular on integral flows, the max-flow min-cut theorem, and Menger's theorem, we refer to the work of Kleinberg and Tardos [19].

Parameterized Complexity. A *parameterized problem* is a set of instances (\mathcal{I}, ℓ) , where $\mathcal{I} \in \Sigma^*$ for a finite alphabet Σ , and $\ell \in \mathbb{N}$ is the *parameter*. A parameterized problem Q is *fixed-parameter tractable*, shortly FPT, if there exists an algorithm that on input (\mathcal{I}, ℓ) decides whether (\mathcal{I}, ℓ) is a yes-instance of Q in $f(\ell) \cdot |\mathcal{I}|^{O(1)}$ time, where f is a computable function independent of $|\mathcal{I}|$.

$W[t]$, $t \geq 1$, are classes that (amongst others) contain parameterized problems which presumably do not admit FPT algorithms. Hardness for $W[t]$ can be shown by reducing from a $W[t]$ -hard problem, using a *parameterized reduction*, that is, a many-to-one reduction that runs in FPT time and maps any instance (\mathcal{I}, ℓ) to another instance (\mathcal{I}', ℓ') such that $\ell' \leq f(\ell)$ for some computable function f .

A parameterized problem Q is *kernelizable* if there exists a polynomial-time self-reduction that maps an instance (\mathcal{I}, ℓ) of Q to another instance (\mathcal{I}', ℓ') of Q such that: (1) $|\mathcal{I}'| \leq \lambda(\ell)$ for some computable function λ , (2) $\ell' \leq \lambda(\ell)$, and (3) (\mathcal{I}, ℓ) is a yes-instance of Q if and only if (\mathcal{I}', ℓ') is a yes-instance of Q . The instance (\mathcal{I}', ℓ') is called the *problem kernel* of (\mathcal{I}, ℓ) and λ is called its *size*.

3 NP-Completeness and $W[2]$ -Hardness With Respect to the Number of Shared Edges

In this section, we show that MINIMUM SHARED EDGES is NP-complete and $W[2]$ -hard with respect to the number k of shared edges. To this end, we give a parameterized, polynomial reduction from the SET COVER problem.

Theorem 1. MINIMUM SHARED EDGES is NP-complete and $W[2]$ -hard with respect to the number k of shared edges.

In the proof of Theorem 1 we provide a reduction from the following problem.

SET COVER (SC)

Input: A set X , a set of sets $\mathcal{C} \subseteq 2^X$, and an integer $\ell \in \mathbb{N}_0$.

Question: Are there sets $C_1, \dots, C_{\ell'} \in \mathcal{C}$ with $\ell' \leq \ell$ such that $X = \bigcup_{i=1}^{\ell'} C_i$?

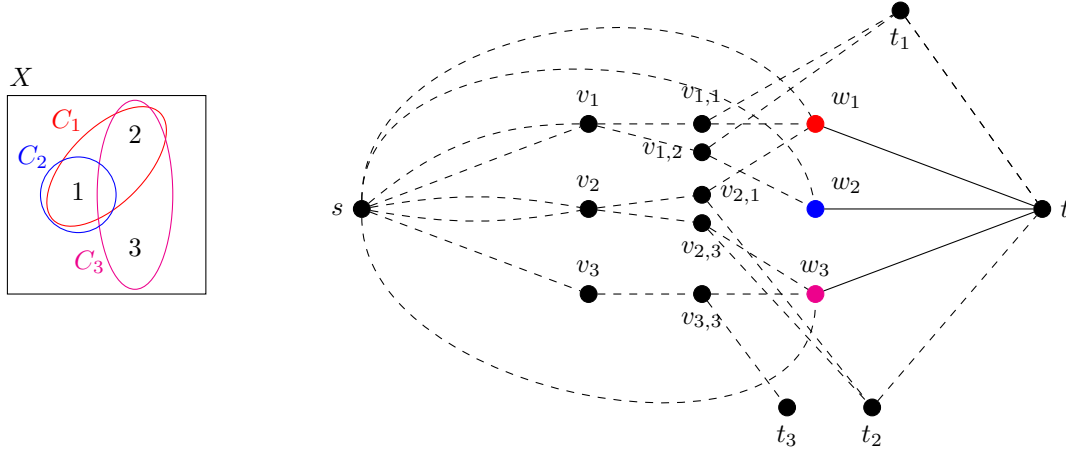


Figure 1: Illustration of the construction of the graph G (right-hand side) in the reduction from an instance of SET COVER on the left-hand side to an instance of MINIMUM SHARED EDGES. Dashed lines represent $(\ell + 1)$ -chains, where ℓ is the parameter in the instance of SET COVER.

SC parameterized by ℓ is well-known to be W[2]-complete [10]. Omran et al. [24] showed that MINIMUM SHARED EDGES on directed graphs is NP-hard using a reduction from SET COVER. Since their reduction is also a parameterized reduction from SC parameterized by ℓ to MINIMUM SHARED EDGES on directed graphs parameterized by the number k of shared edges, they showed implicitly that MINIMUM SHARED EDGES on directed graphs is W[2]-hard with respect to k .

Here, we present a parameterized reduction from SC parameterized by ℓ to MINIMUM SHARED EDGES (on *undirected* graphs) parameterized by k . In the following, we call a path of length $m \in \mathbb{N}$ an m -chain.

Proof of Theorem 1. Let (X, \mathcal{C}, ℓ) be an instance of SC. Let $\deg(x)$ be the number of sets in \mathcal{C} containing element $x \in X$, that is, $\deg(x) := |\{C \in \mathcal{C} \mid x \in C\}|$ for every $x \in X$. We construct an instance (G, s, t, p, k) of MINIMUM SHARED EDGES with $p = |\mathcal{C}| + \sum_{x \in X} \deg(x)$ and $k = \ell$ as follows.

Construction. The construction is illustrated in Figure 1. Initially, let G be an empty graph, that is $V(G) = E(G) = \emptyset$. Next, we add the vertices s and t to the vertex set $V(G)$ of graph G . Then, we add to $V(G)$ the following vertex sets:

- $V_X = \{v_i \mid i \in X\}$, the set of vertices corresponding to the elements of X ,
- $V_C = \{w_j \mid C_j \in \mathcal{C}\}$, the set of vertices corresponding to the sets in \mathcal{C} ,
- $V_D = \{v_{i,j} \mid (i \in X) \wedge (C_j \in \mathcal{C}) \wedge (i \in C_j)\}$, the set of vertices corresponding to the relation of the elements in X with the sets in \mathcal{C} , i.e. a vertex $v_{i,j}$ is in V_D if there is an element $i \in X$ and a set $C_j \in \mathcal{C}$ such that $i \in C_j$, and
- $V_T = \{t_i \mid i \in X\}$.

We connect each $v_{i,j} \in V_D$ via an $(\ell + 1)$ -chain with $v_i \in V_X$, with $t_i \in V_T$ and with $w_j \in V_C$. Next, we connect vertex s with every vertex $w \in V_C$ via an $(\ell + 1)$ -chain and with each $v_i \in V_X$ via $\deg(i)$ $(\ell + 1)$ -chains. Finally, we connect vertex t with each $w \in V_C$ via a single edge each and with each $t_i \in V_T$ via $\deg(i) - 1$ many $(\ell + 1)$ -chains.

Correctness. We now prove that (X, \mathcal{C}, ℓ) is a yes-instance for SC if and only if (G, s, t, p, k) is a yes-instance for MINIMUM SHARED EDGES.

“ \Leftarrow ”: Suppose that we have p s - t routes in G that share at most k edges. We show that we can construct a set cover $\mathcal{C}' \subseteq \mathcal{C}$ for X with $|\mathcal{C}'| \leq \ell$. First, we provide some observations.

Since every $(\ell + 1)$ -chain contains $\ell + 1$ edges, every $(\ell + 1)$ -chain in G appears in at most one s - t route. Since there are p s - t routes and there are p $(\ell + 1)$ -chains incident with vertex s , every $(\ell + 1)$ -chain incident with vertex s appears in exactly one s - t route. Therefore, each $v_i \in V_X$ appears in at least $\deg(i)$ s - t routes and each $w_j \in V_C$ appears in at least one s - t route. Moreover, since each $v_i \in V_X$ is incident with $2 \cdot \deg(i)$ $(\ell + 1)$ -chains, each $v_i \in V_X$ appears in exactly $\deg(i)$ s - t routes.

Each $v_{i,j} \in V_D$ has exactly degree three and is incident with three $(\ell + 1)$ -chains. Therefore, every $v_{i,j} \in V_D$ appears in at most one s - t route. Moreover, since each $v_i \in V_X$ appears in $\deg(i)$ s - t routes, and there are $\deg(i)$ vertices in V_D each connected with v_i via an $(\ell + 1)$ -chain, each $v_{i,j} \in V_D$ appears in exactly one s - t route.

Let $V' := \{w \in V_C \mid \{w, t\} \text{ is a shared edge}\}$, that is, V' is the set of vertices in V_C that are incident with the shared edges of the p s - t routes. We claim that, if $w_j \in V_C$ appears in an s - t route P containing a vertex in V_X , then $w_j \in V'$. Let $v_i \in V_X$ be the vertex that appears in route P . Suppose V' does not contain vertex w_j and, thus, edge $\{w_j, t\}$ is not shared. Since the $(\ell + 1)$ -chain connecting vertex s with vertex w_j appears in exactly one s - t route different from P , vertex w_j appears in at least two s - t routes. Since every vertex in V_C is incident with vertex t and vertices in V_D via $(\ell + 1)$ -chains, there is a vertex $v_{i',j'} \in V_D$ different from vertex $v_{i,j}$, such that one of the s - t routes containing vertex w_j contains vertex $v_{i',j'}$. Let P' be the route containing the vertices w_j and $v_{i',j'}$. We know that there is an s - t route containing vertex $v_{i',j'}$ and vertex $v_{i'}$ different from P' , since there are p s - t routes. Thus, vertex $v_{i',j'}$ appears in at least two s - t routes, contradicting the fact that each vertex in V_D appears in exactly one s - t route. We conclude that set V' contains vertex w_j .

We claim that the subset $\mathcal{C}' \subseteq \mathcal{C}$ corresponding to vertices in V' , that is $\mathcal{C}' := \{C_j \in \mathcal{C} \mid w_j \in V'\}$, is a set cover of X of size at most ℓ . Each $t_i \in V_T$ is connected with vertex t via $\deg(i) - 1$ $(\ell + 1)$ -chains, and connected with $\deg(i)$ vertices in V_D . Therefore, for each $i \in X$, there exists at least one $j \in [|\mathcal{C}|]$, such that $v_i, v_{i,j}$, and w_j appear in an s - t route. As shown before, it follows that $w_j \in V'$. Thus, for each element $i \in X$ there exists a set $C_j \in \mathcal{C}'$ such that $i \in C_j$, and hence, \mathcal{C}' is a set cover of X of size at most ℓ .

" \Rightarrow ": Suppose that we have a set $\mathcal{C}' \subseteq \mathcal{C}$ with $|\mathcal{C}'| \leq \ell$, such that \mathcal{C}' is a set cover of X . We show that we can construct p s - t routes in G that share at most $k = \ell$ edges.

First, we construct $|\mathcal{C}|$ s - t routes in the following way. For each vertex $w \in V_C$, we construct the s - t route containing only the $(\ell + 1)$ -chain connecting s and w and the edge $\{w, t\}$. It follows that each of the $|\mathcal{C}|$ edges connecting a vertex in V_C with vertex t appears in exactly one s - t route.

Next, we construct $|X|$ s - t routes in the following way. We remark that since \mathcal{C}' is a set cover of X , for each $i \in X$ there exists a $C_j \in \mathcal{C}'$ such that $i \in C_j$. For each $v_i \in V_X$, we construct an s - t route containing only the $(\ell + 1)$ -chains connecting s with v_i , v_i with $v_{i,j}$, $v_{i,j}$ with w_j , and the edge $\{w_j, t\}$, where vertex $w_j \in V_C$ corresponds to a $C_j \in \mathcal{C}'$ with $i \in C_j$. Since $|\mathcal{C}'| \leq \ell$, there are at most ℓ edges connecting the vertices in V_C with t that are shared by the s - t routes constructed so far.

Finally, we construct $\sum_{x \in X} \deg(x) - |X|$ s - t routes in the following way. Note that for each $v_i \in V_X$, there are $\deg(i) - 1$ $(\ell + 1)$ -chains connecting s and v_i not covered by an s - t route and there are $\deg(i) - 1$ vertices in V_D connected with v_i via an $(\ell + 1)$ -chain not covered by an s - t route. Moreover, no vertex in V_T is covered by an s - t route, and thus, $t_i \in V_T$ is not covered by an s - t route. Recall that $t_i \in V_T$ is connected with vertex t by $\deg(i) - 1$ $(\ell + 1)$ -chains. Thus, for each $v_i \in V_X$, we can lead $\deg(i) - 1$ s - t routes from s over v_i , vertices in V_D and t_i to t without sharing any edge.

In total, we constructed

$$|\mathcal{C}| + |X| + \sum_{x \in X} \deg(x) - |X| = |\mathcal{C}| + \sum_{x \in X} \deg(x) = p$$

s - t routes sharing at most $k = \ell$ edges.

Note that the reduction is a polynomial reduction and a parameterized reduction since $k = \ell$. Moreover, given p s - t routes in a graph G with $s, t \in V(G)$, one can check in polynomial time whether the routes share at most k edges. Hence, MINIMUM SHARED EDGES is NP-complete and W[2]-hard with respect to the number k of shared edges. \square

4 An Algorithm for Small Treewidth and Small Number of Routes

In this section we show the following theorem.

Theorem 2. *Let G be a graph with $s, t \in V(G)$ given together with a tree decomposition of width ω . Let $p \in \mathbb{N}$ be an integer. Then the minimum number of shared edges in a (p, s, t) -routing can be computed in $O(p \cdot (\omega + 4)^{3 \cdot p \cdot (\omega + 3) + 4} \cdot n)$ time.*

The proof is based on a dynamic program that computes a table for each node of the (arbitrarily rooted) tree decomposition in a bottom-up fashion. For our application, it is convenient to use a nice tree decomposition with introduce edge nodes such that each bag contains the sink and the source node. For each node α in the tree decomposition \mathbb{T} of G , we define V_α as the set of vertices and E_α as the set of edges that are introduced in the subtree rooted at node α . In other words, a vertex $v \in V(G)$ is in V_α if and only if there exists at least one introduce vertex node in the subtree rooted at node α that introduced vertex v . As a special case, since the vertices s and t are contained in every bag, we consider s and t as introduced by each leaf node. An edge $e \in E(G)$ is in E_α if and only if there exists an introduce edge node in the subtree rooted at node α that introduced edge e . Recall that there is a unique introduce edge node for every edge of graph G . We define $G_\alpha := (V_\alpha, E_\alpha)$ as the graph for node α . For every leaf node α in \mathbb{T} , we set $V_\alpha = \{s, t\}$ and $E_\alpha = \emptyset$.

In Figure 2, we show for an example graph G (upper-left) with $s, t \in V(G)$ a nice tree decomposition with introduce edge nodes and vertices s and t contained in each bag. Moreover, we illustrate the graphs as defined above for some tagged nodes in the tree decomposition. In the center of the figure, the modified nice tree decomposition with introduce edge nodes is shown. The graphs around the tree decomposition are the graphs for some tagged nodes, for example, the graph G_α is the graph for node α in the tree decomposition. For following examples and illustrations, we make use of this example throughout this section, and thus, we denote by \mathbb{T}^* the tree decomposition in Figure 2.

Partial Solutions. We define a set of p forests in G_α as a *partial solution* L_α for node α . Instead of asking for p s - t routes that share at most k edges, we can ask for p s - t forests that share at most k edges, where an s - t forest is a forest that contains at least one tree connecting vertices s and t . Note that every forest that contains a tree containing both vertices s and t can be “reduced” to an s - t path. A partial solution L_α has a cost value $c(L_\alpha)$, which is the number of edges in G_α that appear in at least two of the p forests in L_α .

In order to represent the intersection of the trees in a partial solution with the bag that we are currently considering, we use the following notation. For each node α in the tree decomposition \mathbb{T} of G , we consider p -tuples of pairs $\mathcal{X}^\alpha := (\mathcal{Y}_q^\alpha, Z_q^\alpha)_{q=1, \dots, p}$, where for each $q \in [p]$, $Z_q^\alpha \subseteq B_\alpha$ together with $\mathcal{Y}_q^\alpha \subseteq 2^{B_\alpha}$ is a partition of B_α , that is, (i) $\bigcup_{M \in \mathcal{Y}_q^\alpha} M \cup Z_q^\alpha = B_\alpha$, (ii) for all $X, Y \in \mathcal{Y}_q^\alpha \cup \{Z_q^\alpha\}$ with $X \neq Y$ it holds $X \cap Y = \emptyset$. We say that \mathcal{X}^α is a *signature* for node α . For each $q \in [p]$, we call the pair $(\mathcal{Y}_q^\alpha, Z_q^\alpha)$ a *segmentation* of the vertex set B_α . We write segmentation q instead of segmentation with index q for short. We call each $M \in \mathcal{Y}_q^\alpha$ a *segment* of the segmentation q and we call Z_q^α the *zero-segment* of the segmentation q .

To connect signatures (and segmentations) with the partial solutions that they represent, we use the following notation. We say that the signature \mathcal{X}^α is a *valid* signature for node α if there is a partial solution L_α for node α such that for each $q \in [p]$, the zero-segment Z_q^α is the set of nodes in B_α that do not appear in the forest with index q and for each set $M \in \mathcal{Y}_q^\alpha$, there is a tree S in the forest with index q such that $M = B_\alpha \cap V(S)$. In other words, the sets in \mathcal{Y}_q^α correspond to connected components in the forest with index q of the partial solution. We say that \mathcal{X}^α is a signature *induced* by the partial solution L_α if \mathcal{X}^α is a valid signature for node α and the partial solution L_α validates \mathcal{X}^α . In this case, for each $q \in [p]$, the pair $(\mathcal{Y}_q^\alpha, Z_q^\alpha)$ is an *induced* segmentation. We remark that given \mathcal{X}^α , there can be exactly one, more than one or no partial solution with signature \mathcal{X}^α . Given a partial solution L_α for G_α , there is exactly one signature induced by L_α . Let \mathcal{X}^α be a signature for node α such that there is no partial solution for G_α that induces the signature \mathcal{X}^α , then we say that \mathcal{X}^α is an *invalid* signature.

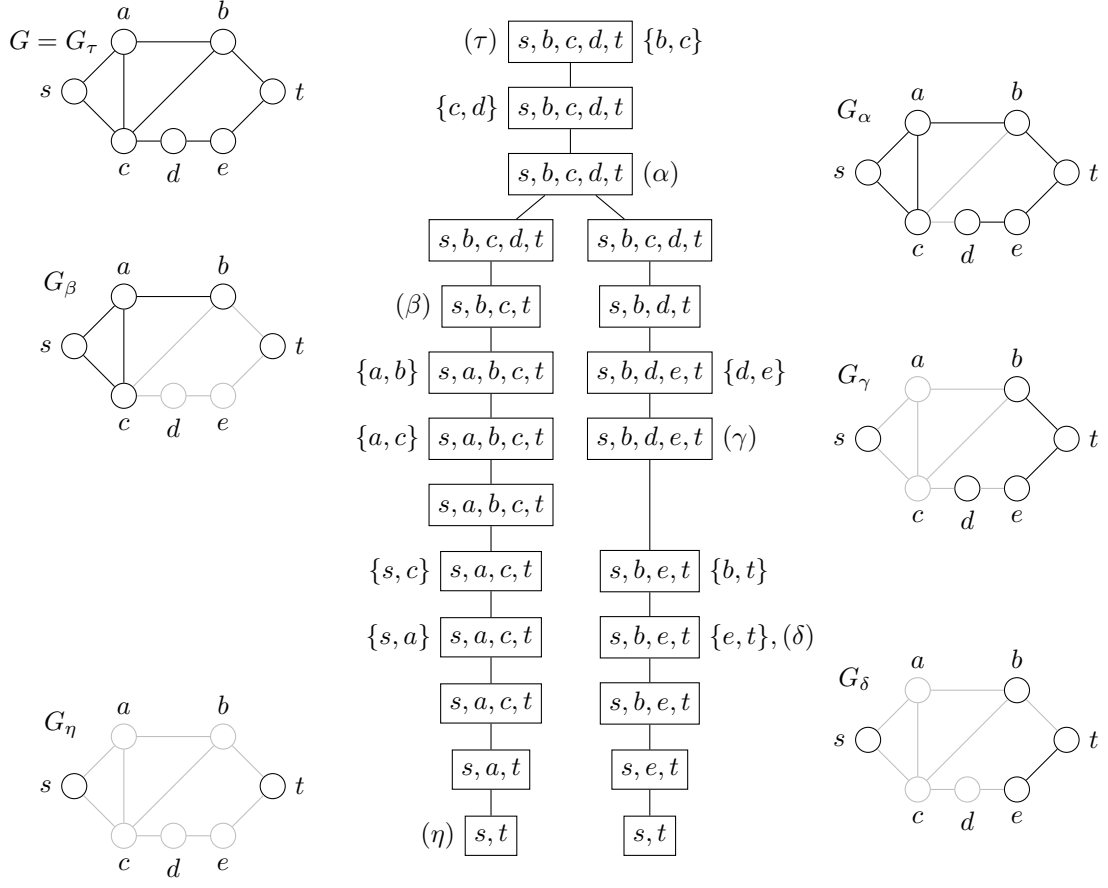


Figure 2: Example for a nice tree decomposition with introduce edge nodes and vertices s and t contained in every bag on an example graph G (top-left). Node τ is the root node in the tree decomposition. The graphs around the tree decomposition correspond to the graphs for the tagged nodes in the tree decomposition. For example, graph G_γ corresponds to the graph for node γ in the tree decomposition.

Given \mathcal{X}^α and $B \subseteq B_\alpha$, we define $\mathcal{X}^\alpha|_B$ as the signature \mathcal{X}^α with sets restricted to the set B , that is, $Z_q^\alpha \cap B$ and $M_q^\alpha \cap B$ for all $M_q^\alpha \in \mathcal{Y}_q^\alpha$ and for all $q \in [p]$.

Let $\mathbb{T} = (T_\mathbb{T}, (B_\alpha)_{\alpha \in V(T_\mathbb{T})})$ be a nice tree decomposition of G with introduce edge nodes and vertices s and t contained in every bag. Let $\omega := \omega(\mathbb{T}) \leq \text{tw}(G) + 2$ be the width of \mathbb{T} . We consider the table T in the following dynamic program that we process bottom-up on the tree decomposition \mathbb{T} , that is, we start to fill the entries of the table T at the leaf nodes of the tree decomposition \mathbb{T} and we traverse the tree of the tree decomposition from the leaves to the root. For a node α in the tree decomposition \mathbb{T} and a signature \mathcal{X}^α for node α , the entry $T[\alpha, \mathcal{X}^\alpha]$ is defined as

$$T[\alpha, \mathcal{X}^\alpha] := \begin{cases} \min c(L_\alpha), & \text{if } \mathcal{X}^\alpha \text{ is a valid signature,} \\ \infty, & \text{otherwise,} \end{cases}$$

where the minimum is taken over all partial solutions L_α in G_α such that L_α induces the signature \mathcal{X}^α .

For each type of node in \mathbb{T} , we define a rule on how to fill each entry in T , prove the correctness of each rule, and discuss the running time for applying the rule and the running time for filling all entries in T for the given type of node. We start with the leaf nodes of the tree decomposition \mathbb{T} .

Leaf Node. Let α be a leaf node of \mathbb{T} . Since s and t appear in every bag of \mathbb{T} , it holds that $B_\alpha = \{s, t\}$. We set

$$T[\alpha, \mathcal{X}^\alpha] := \begin{cases} 0, & \text{if } \mathcal{Y}_q^\alpha = \{\{s\}, \{t\}\} \text{ for all } q = 1, \dots, p, \\ \infty, & \text{otherwise.} \end{cases}$$

We recall that $V_\alpha = \{s, t\}$ and $E_\alpha = \emptyset$ for every leaf node α in \mathbb{T} . Since there is no edge in E_α , the vertices s and t cannot appear together in one tree in a forest in any partial solution in G_α , and thus, the vertices s and t cannot appear together in one segment in any segmentation of a signature for a leaf node. Since in any solution to our problem, s and t appear in each of the p forests, we can set s and t as segments of all p segmentations.

Introduce Vertex Node. Let α be an introduce vertex node of \mathbb{T} and let β be the child node of α with $B_\alpha \setminus B_\beta = \{v\}$. Two signatures \mathcal{X}^α and \mathcal{X}^β are *compatible* if $\mathcal{X}^\alpha|_{B_\beta} = \mathcal{X}^\beta$, and $v \in Z_q^\alpha$ or $\{v\} \in \mathcal{Y}_q^\alpha$ for each $q \in [p]$. We claim that

$$T[\alpha, \mathcal{X}^\alpha] = \begin{cases} \min_{\mathcal{X}^\beta \text{ compatible with } \mathcal{X}^\alpha} T[\beta, \mathcal{X}^\beta], & \text{if it exists } \mathcal{X}^\beta \text{ compatible with } \mathcal{X}^\alpha, \\ \infty, & \text{otherwise.} \end{cases}$$

Since α is an introduce vertex node for vertex $v \in V(G)$, no edge incident with v is introduced in any node in the subtree rooted at node α , and thus, vertex v is an isolated vertex in G_α . As a consequence, in every forest in all partial solutions for G_α , the introduced vertex v is either a single-vertex tree or does not appear in the forest since v cannot be connected to any vertex in G_α . A single-vertex tree is a tree that contains exactly one vertex and does not contain any edge.

Correctness. “ \geq ”: Let L_α be a partial solution for G_α with signature \mathcal{X}^α such that $T[\alpha, \mathcal{X}^\alpha] = c(L_\alpha)$. We construct a partial solution L_β for G_β and a signature \mathcal{X}^β such that L_β induces \mathcal{X}^β and \mathcal{X}^β is compatible with \mathcal{X}^α . For each forest in the partial solution L_α , vertex v is either a single-vertex tree or does not appear in the forest, since there is no edge incident with vertex v in G_α . If vertex v appears as single-vertex tree in any forest in L_α , deleting v yields a forest in G_β . We define the partial solution L_β as L_α restricted to V_β , which are the forests without the isolated vertex v . The partial solution L_β is a partial solution for G_β with valid signature $\mathcal{X}^\beta := \mathcal{X}^\alpha|_{B_\beta}$. Signature \mathcal{X}^β is compatible with signature \mathcal{X}^α . It follows that

$$T[\alpha, \mathcal{X}^\alpha] = c(L_\alpha) = c(L_\beta) \geq T[\beta, \mathcal{X}^\beta] \geq \min_{\mathcal{X}'^\beta \text{ compatible with } \mathcal{X}^\alpha} T[\beta, \mathcal{X}'^\beta].$$

“ \leq ”: Let L_β be a partial solution for G_β with signature \mathcal{X}^β compatible with signature \mathcal{X}^α such that $T[\beta, \mathcal{X}^\beta] = c(L_\beta)$ and $T[\beta, \mathcal{X}^\beta] = \min_{\mathcal{X}'^\beta \text{ compatible with } \mathcal{X}^\alpha} T[\beta, \mathcal{X}'^\beta]$. We construct a partial solution L_α for G_α with signature \mathcal{X}^α . For each $q \in [p]$, if $v \in Z_q^\alpha$, then we do not add v to the forest with index q in L_β . If v is a single segment in the segmentation q , i.e. $\{v\} \in \mathcal{Y}_q^\alpha$, then we add v as a single-vertex tree to the forest with index q in L_β . Since L_β is a partial solution for G_β , the constructed L_α is a partial solution for G_α with signature \mathcal{X}^α . It follows that

$$\min_{\mathcal{X}'^\beta \text{ compatible with } \mathcal{X}^\alpha} T[\beta, \mathcal{X}'^\beta] = T[\beta, \mathcal{X}^\beta] = c(L_\beta) = c(L_\alpha) \geq T[\alpha, \mathcal{X}^\alpha].$$

Running time. For each signature \mathcal{X}^α , we check for all $q \in [p]$ whether $v \in Z_q^\alpha$ or $\{v\} \in \mathcal{Y}_q^\alpha$ in $O(p \cdot |B_\alpha|)$ time. If for all $q \in [p]$ holds that $v \in Z_q^\alpha$ or $\{v\} \in \mathcal{Y}_q^\alpha$, then we check all signatures \mathcal{X}^β for node β for compatibility with signature \mathcal{X}^α , that means, we check if $\mathcal{X}^\alpha|_{B_\beta} = \mathcal{X}^\beta$. This can be done in $O(p \cdot |B_\alpha|^2)$ time. Since there are $O((|B_\beta| + 1)^{p \cdot |B_\beta|})$ signatures for node β and $|B_\beta| \leq |B_\alpha|$, the running time for this step is in $O(p \cdot (|B_\alpha| + 1)^{p \cdot |B_\alpha| + 2})$. Since there are $O((|B_\alpha| + 1)^{p \cdot |B_\alpha|})$ signatures for node α and $|B_\beta| \leq |B_\alpha| \leq \omega + 1$, the overall running time for filling the entries in T for an introduce vertex node is in $O(p \cdot (\omega + 2)^{2 \cdot p \cdot (\omega + 1) + 2})$.

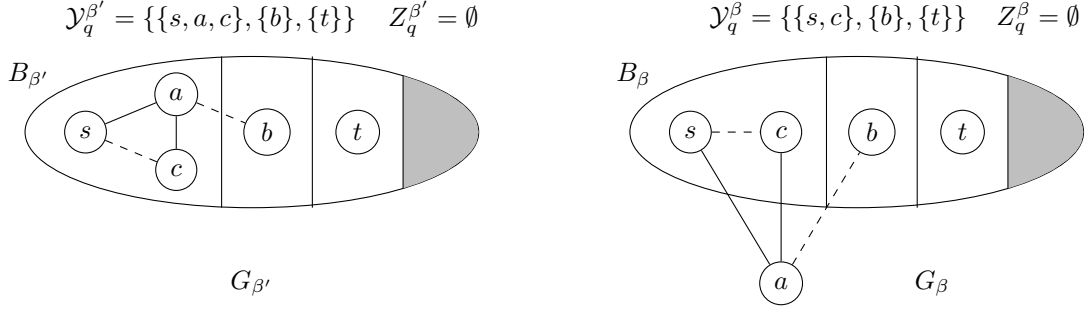


Figure 3: Example for a segmentation q of two compatible signatures \mathcal{X}^{β} and $\mathcal{X}^{\beta'}$ for a forget node β with child node β' in \mathbb{T}^* .

Forget Node. Let α be a forget node of \mathbb{T} and let β be the child node of α with $B_{\beta} \setminus B_{\alpha} = \{v\}$. Two signatures \mathcal{X}^{α} and \mathcal{X}^{β} are *compatible* if $\mathcal{X}^{\alpha} = \mathcal{X}^{\beta}|_{B_{\alpha}}$. We claim that

$$T[\alpha, \mathcal{X}^{\alpha}] = \min_{\mathcal{X}^{\beta} \text{ compatible with } \mathcal{X}^{\alpha}} T[\beta, \mathcal{X}^{\beta}].$$

Since node α is a forget node for the vertex $v \in V(G)$, all edges incident with v have been introduced in the subtree rooted at α . Therefore, every possible way of v appearing in a forest has been considered. We remark that G_{α} and G_{β} are equal.

In Figure 3, we provide an example for a segmentation q of two compatible signatures \mathcal{X}^{β} and $\mathcal{X}^{\beta'}$ for a forget node β with child node β' in \mathbb{T}^* of Figure 2. All lines connecting two vertices correspond to the edges in the graphs $G_{\beta'}$ and G_{β} , where only the solid lines are the edges in the partial solutions that induce the heading segmentations. Node β forgets vertex a . The vertices $s, a, c \in V_{\beta'}$ form a segment in $\mathcal{Y}_q^{\beta'}$. As node β forgets vertex a , the vertices $s, c \in V_{\beta}$ form a segment in \mathcal{Y}_q^{β} , since they are connected via the vertex a .

Correctness. “ \geq ”: Let L_{α} be a partial solution for G_{α} with signature \mathcal{X}^{α} such that $T[\alpha, \mathcal{X}^{\alpha}] = c(L_{\alpha})$. We construct a partial solution L_{β} for G_{β} and a signature \mathcal{X}^{β} such that L_{β} induces \mathcal{X}^{β} and \mathcal{X}^{β} is compatible with \mathcal{X}^{α} . Since $G_{\alpha} = G_{\beta}$, the set of p forests $L_{\beta} := L_{\alpha}$ is a partial solution for G_{β} . We set $\mathcal{X}^{\beta}|_{B_{\alpha}} := \mathcal{X}^{\alpha}$. For each $q \in [p]$, if vertex v does not appear in the forest with index q , then we set $Z_q^{\beta} := Z_q^{\alpha} \cup \{v\}$ and $\mathcal{Y}_q^{\beta} := \mathcal{Y}_q^{\alpha}$. If vertex v appears in the forest with index q , then we set $Z_q^{\beta} := Z_q^{\alpha}$, and we add v to the segmentation \mathcal{Y}_q^{β} as follows. If vertex v appears as a single-vertex tree in the forest with index q , then we add $\{v\}$ to \mathcal{Y}_q^{β} . If vertex v appears in a tree with vertices in $M \in \mathcal{Y}_q^{\alpha}$, then we set $\mathcal{Y}_q^{\beta} := (\mathcal{Y}_q^{\alpha} \setminus \{M\}) \cup \{M \cup \{v\}\}$. Signature \mathcal{X}^{β} is compatible with signature \mathcal{X}^{α} , and the partial solution L_{β} induces \mathcal{X}^{β} . It follows that

$$T[\alpha, \mathcal{X}^{\alpha}] = c(L_{\alpha}) = c(L_{\beta}) \geq T[\beta, \mathcal{X}^{\beta}] \geq \min_{\mathcal{X}^{\beta'} \text{ compatible with } \mathcal{X}^{\alpha}} T[\beta, \mathcal{X}^{\beta'}].$$

“ \leq ”: Let L_{β} be a partial solution for G_{β} with signature \mathcal{X}^{β} compatible with \mathcal{X}^{α} such that $T[\beta, \mathcal{X}^{\beta}] = c(L_{\beta})$ and $T[\beta, \mathcal{X}^{\beta}] = \min_{\mathcal{X}^{\beta'} \text{ compatible with } \mathcal{X}^{\alpha}} T[\beta, \mathcal{X}^{\beta'}]$. We construct a partial solution for G_{α} that induces \mathcal{X}^{α} . Since $G_{\alpha} = G_{\beta}$, we set $L_{\alpha} := L_{\beta}$ as the partial solution L_{α} for G_{α} . Since $\mathcal{X}^{\alpha} = \mathcal{X}^{\beta}|_{B_{\alpha}}$, the partial solution L_{α} induces \mathcal{X}^{α} . It follows that

$$\min_{\mathcal{X}^{\beta'} \text{ compatible with } \mathcal{X}^{\alpha}} T[\beta, \mathcal{X}^{\beta'}] = T[\beta, \mathcal{X}^{\beta}] = c(L_{\beta}) = c(L_{\alpha}) \geq T[\alpha, \mathcal{X}^{\alpha}].$$

Running time. For a signature \mathcal{X}^{α} , we check whether $\mathcal{X}^{\alpha} = \mathcal{X}^{\beta}|_{B_{\alpha}}$ for all signatures \mathcal{X}^{β} for node β . This can be done in $O(p \cdot (|B_{\beta}| + 1)^{p \cdot |B_{\beta}| + 2})$ time. Since $|B_{\alpha}| \leq |B_{\beta}| \leq \omega + 1$, the overall running time for filling all entries in T for a forget node is in $O(p \cdot (\omega + 2)^{2 \cdot p \cdot (\omega + 1) + 2})$.

Introduce Edge Node. Let α be an introduce edge node of \mathbb{T} , let β be the child node of α , and let $e = \{v, w\}$ be the edge introduced by node α . Two signatures \mathcal{X}^{α} and \mathcal{X}^{β} are *compatible* if for each $q \in [p]$, one of the following conditions holds:

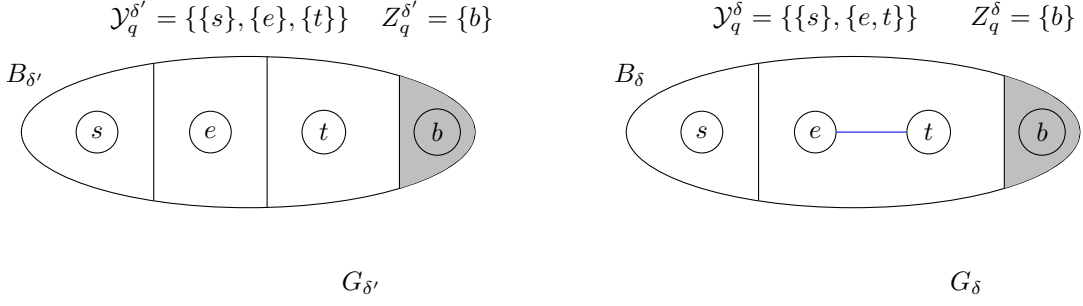


Figure 4: Example for a segmentation q of two compatible signatures \mathcal{X}^{δ} and $\mathcal{X}^{\delta'}$ of an introduce edge node δ with child node δ' in \mathbb{T}^* .

- (i) $\mathcal{Y}_q^{\alpha} = \mathcal{Y}_q^{\beta}$, or
- (ii) $\mathcal{Y}_q^{\alpha} = (\mathcal{Y}_q^{\beta} \setminus \{M_1, M_2\}) \cup \{M_1 \cup M_2\}$ with $M_1, M_2 \in \mathcal{Y}_q^{\beta}$, $M_1 \neq M_2$, and $v \in M_1$ and $w \in M_2$. If \mathcal{X}^{α} and \mathcal{X}^{β} are compatible, then let $Q \subseteq [p]$ be the set of indices such that for all $q \in Q$ (ii) holds and for all $q \in [p] \setminus Q$ (i) holds. We say that \mathcal{X}^{α} and \mathcal{X}^{β} are *share-compatible* if $|Q| \geq 2$. We claim that

$$T[\alpha, \mathcal{X}^{\alpha}] = \min_{\mathcal{X}^{\beta} \text{ compatible with } \mathcal{X}^{\alpha}} \left(T[\beta, \mathcal{X}^{\beta}] + \begin{cases} 1, & \text{if } \mathcal{X}^{\beta} \text{ and } \mathcal{X}^{\alpha} \text{ are share-compatible,} \\ 0, & \text{otherwise.} \end{cases} \right)$$

In other words, two signatures \mathcal{X}^{α} for node α and \mathcal{X}^{β} for node β are compatible if and only if for all $q \in [p]$, either by (i) it holds that the segmentation q in \mathcal{X}^{α} is equal to the segmentation q of \mathcal{X}^{β} , or by (ii) it holds that the segmentation q of \mathcal{X}^{α} is the result of merging two segments in the segmentation q of \mathcal{X}^{β} , where none of the two segments is the zero-segment, and vertex v is in the one segment, and vertex w is in the other segment. This corresponds to connecting two trees by edge e in the forest with index q , where v is in the one tree and w in the other tree. Note that connecting two vertex-disjoint trees by exactly one edge yields a tree. The deletion of edge e in every forest of a partial solution for G_{α} that includes the edge e yields a partial solution for G_{β} . We remark that $G_{\alpha} = G_{\beta} + \{e\}$, that is, G_{α} differs from G_{β} only by the additional edge e . In Figure 4, we provide an example for a segmentation q of two compatible signatures \mathcal{X}^{δ} and $\mathcal{X}^{\delta'}$ of an introduce edge node δ with child node δ' in \mathbb{T}^* . Graph $G_{\delta'}$ does not contain any edge. Edge $\{e, t\}$ is introduced by node δ . This allows to connect the segments containing vertex e on the one hand, and vertex t on the other hand, using edge $\{e, t\}$.

Correctness. “ \geq ”: Let L_{α} be a partial solution for G_{α} with signature \mathcal{X}^{α} such that $T[\alpha, \mathcal{X}^{\alpha}] = c(L_{\alpha})$. We construct a partial solution L_{β} for G_{β} and a signature \mathcal{X}^{β} such that L_{β} induces \mathcal{X}^{β} and \mathcal{X}^{β} is compatible with \mathcal{X}^{α} . For each $q \in [p]$, if edge e is not part of the forest with index q in L_{α} , then the forest is a forest in G_{β} as well. Then, we set $Z_q^{\beta} := Z_q^{\alpha}$ and $\mathcal{Y}_q^{\beta} := \mathcal{Y}_q^{\alpha}$. If edge e is part of the forest with index q in L_{α} , then deleting edge e from the forest with index q disconnects a tree of the forest such that two trees result, with v in the one tree and w in the other tree. Let $M \in \mathcal{Y}_q^{\alpha}$ be the segment in the segmentation q with $v, w \in M$. Let M_1, M_2 be the induced sets by splitting tree T_{α} in the forest with index q in L_{α} at edge e , that is, if T_1 and T_2 are the connected subgraphs of $T_{\alpha} \setminus \{e\}$, then $M_1 := V(T_1) \cap B_{\alpha}$ and $M_2 := V(T_2) \cap B_{\alpha}$. We set $\mathcal{Y}_q^{\beta} := (\mathcal{Y}_q^{\alpha} \setminus \{M\}) \cup \{M_1, M_2\}$ and $Z_q^{\beta} := Z_q^{\alpha}$. Signature \mathcal{X}^{β} for node β is compatible with signature \mathcal{X}^{α} for node α .

Let L_{β} be the set of p forests in L_{α} restricted to edge set E_{β} . Then, L_{β} is a partial solution for G_{β} and induces signature \mathcal{X}^{β} . If edge e appears in more than one of the p forests in L_{α} , then $c(L_{\beta}) = c(L_{\alpha}) - 1$ and the signatures \mathcal{X}^{α} and \mathcal{X}^{β} are share-compatible. It follows that

$$T[\alpha, \mathcal{X}^{\alpha}] = c(L_{\alpha}) = c(L_{\beta}) + 1 \geq T[\beta, \mathcal{X}^{\beta}] + 1 \geq \min_{\mathcal{X}^{\beta'} \text{ compatible with } \mathcal{X}^{\alpha}} T[\beta, \mathcal{X}^{\beta'}] + 1.$$

If edge e appears in at most one of the p forests in the partial solution L_{α} , then

$$T[\alpha, \mathcal{X}^{\alpha}] = c(L_{\alpha}) = c(L_{\beta}) \geq T[\beta, \mathcal{X}^{\beta}] \geq \min_{\mathcal{X}^{\beta'} \text{ compatible with } \mathcal{X}^{\alpha}} T[\beta, \mathcal{X}^{\beta'}].$$

“ \leq ”: Let L_β be a partial solution for G_β with signature \mathcal{X}^β compatible with \mathcal{X}^α such that $T[\beta, \mathcal{X}^\beta] = c(L_\beta)$ and $T[\beta, \mathcal{X}^\beta] = \min_{\mathcal{X}'^\beta \text{ compatible with } \mathcal{X}^\alpha} T[\beta, \mathcal{X}'^\beta]$. We construct a partial solution L_α for G_α that induces signature \mathcal{X}^α . For each $q \in [p]$, if condition (i) holds, that is, if $\mathcal{Y}_q^\alpha = \mathcal{Y}_q^\beta$, then we set the forest with index q in L_α to the forest with index q in the partial solution L_β . In case of condition (ii), that is, if v and w belong to the same segment in the segmentation q of \mathcal{X}^α but are not in the same segment in the segmentation q of \mathcal{X}^β , then we add edge e to the forest with index q in the partial solution L_β and we set the resulting forest as the forest with index q in the partial solution L_α . Since the vertices v and w are in two vertex-disjoint trees in the forest with index q in L_β , adding edge e connects the two trees at the vertices v and w , which results again in a tree. The set of p forests L_α , constructed as mentioned above, is a partial solution for G_α and induces signature \mathcal{X}^α . If the signatures \mathcal{X}^β and \mathcal{X}^α are share-compatible, then the partial solution L_α is the result of adding edge e to at least two forests in L_β , and thus, the number of common edges of the forests increases by exactly one, i.e. $c(L_\beta) = c(L_\alpha) - 1$. It follows that

$$\min_{\mathcal{X}'^\beta \text{ compatible with } \mathcal{X}^\alpha} T[\beta, \mathcal{X}'^\beta] = T[\beta, \mathcal{X}^\beta] = c(L_\beta) = c(L_\alpha) - 1 \geq T[\alpha, \mathcal{X}^\alpha] - 1.$$

If the signatures \mathcal{X}^β and \mathcal{X}^α are compatible but not share-compatible, then the partial solution L_α is the result of adding edge e to at most one forest in L_β . It follows that

$$\min_{\mathcal{X}'^\beta \text{ compatible with } \mathcal{X}^\alpha} T[\beta, \mathcal{X}'^\beta] = T[\beta, \mathcal{X}^\beta] = c(L_\beta) = c(L_\alpha) \geq T[\alpha, \mathcal{X}^\alpha].$$

Running time. For each signature \mathcal{X}^α , we check all signatures \mathcal{X}^β for node β for compatibility, that means, we need to check for each $q \in [p]$ whether the segmentations are equal (i) or whether the segmentation q of \mathcal{X}^α is derived by merging two segments in the segmentation q of \mathcal{X}^β (ii). To check condition (i) as well as to check condition (ii) can be done in $O(p \cdot |B_\alpha|^2)$ time. Therefore, the overall running time for filling all entries in T for an introduce edge node is in $O(p \cdot (\omega + 2)^{2 \cdot p \cdot (\omega + 1) + 2})$.

Join Node. Let α be a join node of \mathbb{T} and let β, γ be the two child nodes of α . A signature \mathcal{X}^α for node α and a pair of two signatures \mathcal{X}^β for node β and \mathcal{X}^γ for node γ are *compatible* if for all $q \in [p]$ it holds that

- (i) $Z_q^\alpha = Z_q^\beta = Z_q^\gamma$,
- (ii) $v, w \in M^\alpha \in \mathcal{Y}_q^\alpha$ with $v \neq w$ if and only if there exists $\ell \geq 1$ and $M_1, \dots, M_\ell \in \mathcal{Y}_q^\beta \cup \mathcal{Y}_q^\gamma$ with $|M_i \cap M_{i+1}| = 1$ for all $i = 1, \dots, \ell - 1$ and $v \in M_1$ and $w \in M_\ell$,
- (iii) for all $M^\beta \in \mathcal{Y}_q^\beta$ and $M^\gamma \in \mathcal{Y}_q^\gamma$ holds $|M^\beta \cap M^\gamma| \leq 1$, and
- (iv) there do not exist $\ell \geq 3$ and $M_1, \dots, M_\ell \in \mathcal{Y}_q^\beta \cup \mathcal{Y}_q^\gamma$ with $|M_i \cap M_{i+1}| = 1$ for all $i = 1, \dots, \ell - 1$ and $M_i \neq M_j$ for all $i, j \in [\ell]$, $i \neq j$, such that $v \in M_1$ and $w \in M_\ell$.

We claim that

$$T[\alpha, \mathcal{X}^\alpha] = \min_{(\mathcal{X}^\beta, \mathcal{X}^\gamma) \text{ compatible with } \mathcal{X}^\alpha} (T[\beta, \mathcal{X}^\beta] + T[\gamma, \mathcal{X}^\gamma]).$$

In other words, a signature \mathcal{X}^α is compatible with a pair of two signatures \mathcal{X}^β for node β and \mathcal{X}^γ for node γ , if and only if for every $q \in [p]$ it holds that

- (i) the vertices that appear in the segmentations with index q in all three signatures are the same,
- (ii) every segment in the segmentation q of \mathcal{X}^α is a union of segments in the segmentation q in \mathcal{X}^β and segments in the segmentation q in \mathcal{X}^γ ,
- (iii) every pair of segments with one segment in the segmentation q in \mathcal{X}^β and one segment in the segmentation q in \mathcal{X}^γ has at most one vertex in $B_\beta = B_\gamma$ in common, and
- (iv) there is no chain of at least three segments in the union of the segmentations with index q in \mathcal{X}^β and \mathcal{X}^γ with one vertex in the first and last segment.

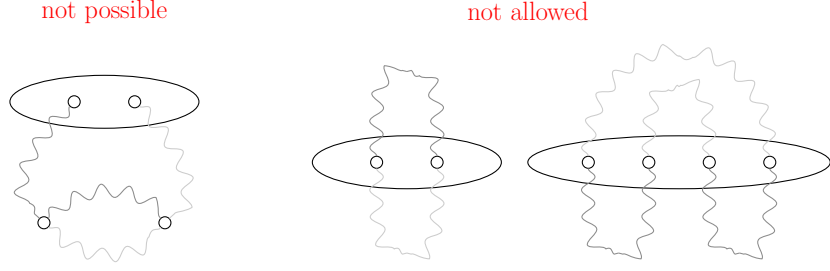


Figure 5: Sketch of scenarios when combining two forests with the same index in two partial solutions for the two child nodes of a join node.

We say that segments M_1, \dots, M_ℓ , $\ell \geq 2$, form a *chain* of segments, if $|M_i \cap M_{i+1}| = 1$ for all $i = 1, \dots, \ell - 1$ and $M_i \neq M_j$ for all $i, j \in [\ell]$, $i \neq j$.

Intuitively, (i)-(iv) define how to combine segmentations in signatures of child nodes to segmentations in a signature of a join node. The condition (ii) ensures that every forest with index q in a partial solution L_α for G_α is a union of the two forests with index q in some partial solutions L_β for G_β and L_γ for G_γ , respectively.

Figure 5 exemplifies three scenarios of cycle creation caused by a union of two forests, where the colors dark-gray and light-gray indicate each of the two forests. We used curved lines to highlight that there could be trees connecting two vertices. The scenario on the left-hand side illustrates a situation that is not possible. The scenario implies that there are vertices that have been forgotten in the subtrees rooted at each child node of the tree decomposition, which is not possible by the definition of a tree decomposition (cf. Section 2). The two scenarios on the right-hand side illustrate two scenarios that are not allowed to occur by our definition of compatibility of join nodes. More precisely, conditions (iii) and (iv) ensure that none of these two scenarios occurs.

The conditions (iii) and (iv) ensure that a union of two forests in L_β and L_γ does not close a cycle. Condition (iii) prevents the following creation of cycles. If there is a tree T_β in the forest with index q in L_β and a tree T_γ in the forest with index q in L_γ that have at least two vertices in common, then the union of these two trees creates a cycle in G_α . Condition (iv) prevents the following creation of cycles. Let v be a vertex in V_α such that there exist some trees T_1, \dots, T_ℓ in the forests with index q in L_β and L_γ such that $|V(T_i) \cap V(T_{i+1})| = 1$ for $i = 1, \dots, \ell - 1$ and $v \in V(T_1)$ and $v \in V(T_\ell)$. Then the graph $T^\alpha = T_1 \cup \dots \cup T_\ell$ as union of the trees in G_α contains a cycle and vertex v is part of a cycle in T^α .

In Figure 6, we provide an example for a segmentation q of three compatible signatures \mathcal{X}^α , $\mathcal{X}^{\alpha_\ell}$, and \mathcal{X}^{α_r} for a join node α with child nodes α_ℓ and α_r in \mathbb{T}^* . The segments $\{s, b\}$ and $\{t\}$ in $\mathcal{Y}_q^{\alpha_\ell}$ together with the segments $\{s\}$ and $\{b, t\}$ in \mathcal{Y}^{α_r} form segment $\{s, b, t\}$ in \mathcal{Y}_q^α . Note that the four conditions for compatibility hold. Condition (i) holds since $Z_q^\alpha = Z_q^{\alpha_\ell} = Z_q^{\alpha_r} = \{d\}$. Moreover, note that condition (iii) holds. According to condition (ii), note that for any pair in the segment $\{s, b, t\} \in \mathcal{Y}_q^\alpha$, the segments $\{s, b\} \in \mathcal{Y}_q^{\alpha_\ell}$ and $\{b, t\} \in \mathcal{Y}_q^{\alpha_r}$ provide a required chain of segments. Conversely, for any possible chain of segments in $\mathcal{Y}_q^{\alpha_\ell} \cup \mathcal{Y}_q^{\alpha_r}$, segment $\{s, b, t\} \in \mathcal{Y}_q^\alpha$ is the required segment in condition (ii). According to condition (iv), note that there is no chain of at least three segments in $\mathcal{Y}_q^{\alpha_\ell} \cup \mathcal{Y}_q^{\alpha_r}$ such that a vertex $v \in \{s, b, c, t\}$ appears in the first and last segment of the chain.

Correctness. “ \geq ”: Let L_α be a partial solution for G_α with signature \mathcal{X}^α such that $T[\alpha, \mathcal{X}^\alpha] = c(L_\alpha)$. We construct a partial solution L_β for G_β , a partial solution L_γ for G_γ and two signatures \mathcal{X}^β and \mathcal{X}^γ , such that the pair $(\mathcal{X}^\beta, \mathcal{X}^\gamma)$ is compatible with \mathcal{X}^α , the partial solution L_β induces signature \mathcal{X}^β and the partial solution L_γ induces signature \mathcal{X}^γ . If we restrict each forest in L_α to the edge sets E_β and E_γ , then each forest restricted to E_β is a forest in G_β and each forest restricted to E_γ is a forest in G_γ . Therefore, restricting each forest in L_α to E_β yields a partial solution L_β for G_β , and restricting each forest in L_α to E_γ yields a partial solution L_γ for G_γ . We set \mathcal{X}^β and \mathcal{X}^γ as the signatures induced by the partial solutions L_β and L_γ respectively.

We show that the pair of signatures \mathcal{X}^β and \mathcal{X}^γ is compatible with \mathcal{X}^α . Condition (i) holds for every $q \in [p]$ since every vertex that does not appear in the forest with index q in L_α neither appears in the forests with index q nor in L_β nor in L_γ . Since the segmentations with index q are

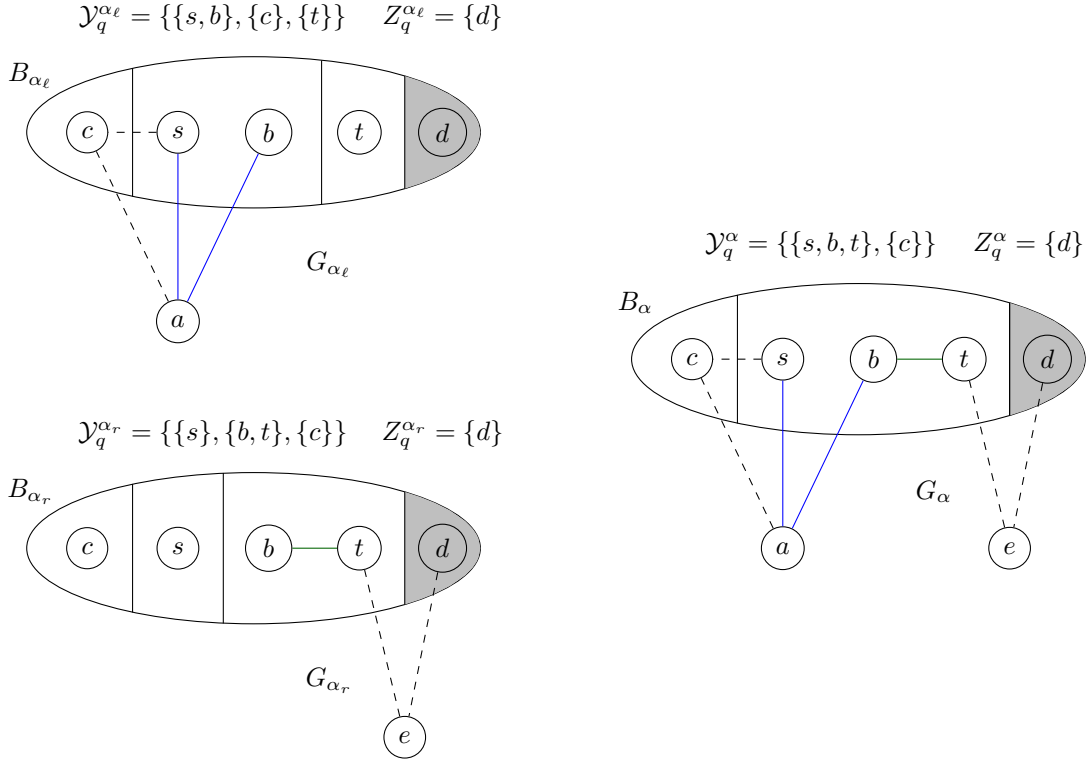


Figure 6: Example for a segmentation q of three compatible signatures \mathcal{X}^α , $\mathcal{X}^{\alpha_\ell}$, and \mathcal{X}^{α_r} for a join node α with child nodes α_ℓ and α_r in \mathbb{T}^* .

induced by L_β and L_γ , it follows that $Z_q^\alpha = Z_q^\beta = Z_q^\gamma$ for all $q \in [p]$.

Suppose that there exists a $q \in [p]$ such that condition (iii) does not hold for $q \in [p]$. This means that there exist $M^\beta \in \mathcal{Y}_q^\beta$ and $M^\gamma \in \mathcal{Y}_q^\gamma$ with $|M^\beta \cap M^\gamma| \geq 2$. Let $v, w \in M^\beta \cap M^\gamma$. Let T^β be the tree in the forest with index q in L_β corresponding to M^β and let T^γ be the tree in the forest with index q in L_γ corresponding to M^γ . Note that $v, w \in V(T^\beta) \cap V(T^\gamma)$. By our construction of L_β and L_γ , there is a tree T^α in the forest with index q in L_α , such that T^β is a subtree of T^α restricted to E_β , and T^γ is a subtree of T^α restricted to E_γ . Since $v, w \in V(T^\alpha)$, there is a v - w path in T^α using only edges in E_β and a v - w path in T^α using only edges in E_γ . Since $E_\beta \cap E_\gamma = \emptyset$, the two paths form a cycle in T^α . This is a contradiction to the fact that T^α is a tree.

For condition (ii), direction “ \Rightarrow ”, we consider $q \in [p]$, $M^\alpha \in \mathcal{Y}_q^\alpha$ and $v, w \in M^\alpha$, $v \neq w$, if such a $M^\alpha \in \mathcal{Y}_q^\alpha$ exists. Segment M^α corresponds to a tree T^α in the forest with index q in L_α . Since $v, w \in M^\alpha$, the vertices v and w appear in tree T^α . Since $E_\alpha = E_\beta \cup E_\gamma$ and $E_\beta \cap E_\gamma = \emptyset$, the restriction of T^α to E_β and E_γ splits the tree in maximal subtrees T_1, \dots, T_ℓ alternating by G_β and G_γ . Note that $|V(T_i) \cap V(T_j)| \leq 1$ for all $i, j \in [\ell]$, $i \neq j$, and $T^\alpha = T_1 \cup \dots \cup T_\ell$. Let $M_1, \dots, M_\ell \in \mathcal{Y}_q^\beta \cup \mathcal{Y}_q^\gamma$ be segments such that segment M_i corresponds to subtree T_i for all $i \in [\ell]$. We claim that if $|V(T_i) \cap V(T_j)| = 1$ for some $i \neq j$ and $u \in V(T_i) \cap V(T_j)$, then $u \in B_\alpha$.

Suppose that $u \notin B_\alpha = B_\beta = B_\gamma$. Since the trees T_1, \dots, T_ℓ are maximal subtrees of tree T^α restricted to E_β and E_γ , one of the trees T_i or T_j is a tree in G_β , and the other is a tree in G_γ . Therefore, vertex u is incident with an edge in E_β and an edge in E_γ . Thus, vertex u appears in the subtree rooted at node β and in the subtree rooted at node γ . This is a contradiction to the fact that \mathbb{T} is a tree decomposition, and hence, $u \in B_\alpha = B_\beta = B_\gamma$.

Moreover, if $|V(T_i) \cap V(T_j)| = 1$ for some $i \neq j$ and $u \in V(T_i) \cap V(T_j)$, then $u \in M_i$ and $u \in M_j$. If there is a $j \in [\ell]$ such that $v, w \in M_j$, then we are done. Thus, let $v \in M_{j_1}$ and $w \in M_{j_2}$ with $j_1, j_2 \in [\ell]$, $j_1 \neq j_2$. Then there exists a subset $S_1, \dots, S_{\ell'}$ of the trees T_1, \dots, T_ℓ with $\ell' \leq \ell$, $S_1 = T_{j_1}$, $S_{\ell'} = T_{j_2}$ and $|V(S_i) \cap V(S_{i+1})| = 1$ for all $i = 1, \dots, \ell' - 1$. Let $M_{S_1}, \dots, M_{S_{\ell'}}$ be the corresponding segments to $S_1, \dots, S_{\ell'}$. Then, $|M_{S_i} \cap M_{S_{i+1}}| = 1$ for all $i = 1, \dots, \ell' - 1$, $v \in M_{S_1}$

and $w \in M_{S'}$, and hence, direction “ \Rightarrow ” of condition (ii) is proven.

For condition (ii), direction “ \Leftarrow ”, we consider $q \in [p]$, $\ell \geq 1$ and $M_1, \dots, M_\ell \in \mathcal{Y}_q^\beta \cup \mathcal{Y}_q^\gamma$ with $|M_i \cap M_{i+1}| = 1$ for all $i = 1, \dots, \ell - 1$, $v \in M_1$ and $w \in M_\ell$. We show that there exists a segment $M^\alpha \in \mathcal{Y}_q^\alpha$ with $v, w \in M^\alpha$. Let T_1, \dots, T_ℓ be trees in the forests with index q in L_β and L_γ such that tree T_i corresponds to segment M_i for all $i \in [\ell]$. Since $|M_i \cap M_{i+1}| = 1$ for all $i = 1, \dots, \ell - 1$, it follows that $|V(T_i) \cap V(T_{i+1})| = 1$ for all $i = 1, \dots, \ell - 1$. Therefore, T_1, \dots, T_ℓ are subtrees of a tree T^α in the forest with index q in L_α with $v, w \in V(T^\alpha)$. Let M^α be the segment corresponding to T^α . Then, segment M^α contains the vertices v and w , i.e. $v, w \in M^\alpha$, and hence, direction “ \Leftarrow ” of condition (ii) is proven.

Suppose that there exists a $q \in [p]$ such that condition (iv) does not hold for $q \in [p]$. Then there exist a vertex $v \in B_\alpha$, an integer $\ell \geq 3$ and segments $M_1, \dots, M_\ell \in \mathcal{Y}_q^\beta \cup \mathcal{Y}_q^\gamma$ with $|M_i \cap M_{i+1}| = 1$ for all $i = 1, \dots, \ell - 1$ and $M_i \neq M_j$ for all $i \neq j$, such that $v \in M_1$ and $v \in M_\ell$. Let T_1, \dots, T_ℓ be the trees in the forests with index q in L_β and L_γ such that tree T_i corresponds to segment M_i for all $i \in [\ell]$. Note that $|V(T_i) \cap V(T_{i+1})| = 1$ for all $i = 1, \dots, \ell - 1$, and vertex v appears in the trees T_1 and T_ℓ . For all $i = 1, \dots, \ell - 1$, let w_i be the vertex in the intersection $V(T_i) \cap V(T_{i+1})$ of the vertex sets of the trees T_i and T_{i+1} . By construction, the union of the trees $T' := T_1 \cup \dots \cup T_\ell$ is a subtree of a tree T^α in the forest with index q in L_α . Thus, the tuple $(v, w_1, w_2, \dots, w_{\ell-1}, v)$ represents a cycle in T' , and thus, in T^α . This is a contradiction to the fact that L_α is a partial solution for G_α , and hence, condition (iv) holds.

We conclude that the pair of signatures \mathcal{X}^β and \mathcal{X}^γ is compatible with \mathcal{X}^α . Since $E_\beta \cap E_\gamma = \emptyset$, the number of edges that appear in at least two forests in L_α is the sum of the number of edges that appear in at least two forests in L_β and the number of edges that appear in at least two forests in L_γ . It follows that

$$\begin{aligned} T[\alpha, \mathcal{X}^\alpha] &= c(L_\alpha) = c(L_\beta) + c(L_\gamma) \geq T[\beta, \mathcal{X}^\beta] + T[\gamma, \mathcal{X}^\gamma] \\ &\geq \min_{(\mathcal{X}'^\beta, \mathcal{X}'^\gamma) \text{ compatible with } \mathcal{X}^\alpha} (T[\beta, \mathcal{X}'^\beta] + T[\gamma, \mathcal{X}'^\gamma]). \end{aligned}$$

“ \leq ”: Let L_β and L_γ be partial solutions for G_β and G_γ with signatures \mathcal{X}^β and \mathcal{X}^γ , as pair compatible with signature \mathcal{X}^α for node α , such that $T[\beta, \mathcal{X}^\beta] = c(L_\beta)$, $T[\gamma, \mathcal{X}^\gamma] = c(L_\gamma)$ and $T[\beta, \mathcal{X}^\beta] + T[\gamma, \mathcal{X}^\gamma] = \min_{(\mathcal{X}'^\beta, \mathcal{X}'^\gamma) \text{ compatible with } \mathcal{X}^\alpha} (T[\beta, \mathcal{X}'^\beta] + T[\gamma, \mathcal{X}'^\gamma])$. We construct a partial solution L_α for G_α with signature \mathcal{X}^α . We claim that for each $q \in [p]$, the union of the forests with index q in L_β and L_γ yields a forest in G_α , that induces the segmentation $(\mathcal{Y}_q^\alpha, Z_q^\alpha)$ in signature \mathcal{X}^α .

Let $B := B_\alpha$. We remark that $B_\alpha = B_\beta = B_\gamma$ since α is a join node in \mathbb{T} . We claim that the intersection of the vertex sets of G_β and G_γ are only the vertices in B , that is $V_\beta \cap V_\gamma = B$. Suppose that there is a vertex $v \in (V_\beta \cap V_\gamma) \setminus B$. Then the graph induced by the node set $\{\rho \in V(\mathbb{T}) \mid v \in B_\rho\}$ is not connected. This contradicts the fact that \mathbb{T} is a tree decomposition, and thus, $V_\beta \cap V_\gamma = B$.

Recall that for each $q \in [p]$, the zero-segments are equal in all three segmentations, that is, $Z_q^\alpha = Z_q^\beta = Z_q^\gamma$. Hence, the vertex sets in both forests with index q in L_β and L_γ are the same. In addition, we know that $E_\beta \cap E_\gamma = \emptyset$ and therefore, the two forests with index q in L_β and L_γ do not have any edge in common. We need to show that for all $q \in [p]$ the union of the forests with index q in L_β and L_γ does not contain a cycle in G_α . Suppose there is a $q \in [p]$ such that the union of the forests with index q in L_β and L_γ contains a cycle in G_α .

Case 1: There is a tree T_1 in the forest with index q in L_β and a tree T_2 in the forest with index q in L_γ , such that the union $T_0 := T_1 \cup T_2$ contains a cycle. Let $M_1 \in \mathcal{Y}_q^\beta$ and $M_2 \in \mathcal{Y}_q^\gamma$, such that segment M_1 corresponds to tree T_1 and segment M_2 corresponds to tree T_2 . Since graph T_0 contains a cycle in G_α , the trees T_1 and T_2 have at least two vertices in common. Because of $V(T_1) \subseteq V_\beta$, $V(T_2) \subseteq V_\gamma$ and $V_\beta \cap V_\gamma = B$, the common vertices are in the vertex set B . This means that there are two vertices $v, w \in B$ such that $v, w \in M_1$ and $v, w \in M_2$. This contradicts condition (iii), and hence, there are no two trees in the forests with index q in L_β and L_γ such that their union contains a cycle in G_α .

Case 2: There are trees T_1, \dots, T_ℓ , $\ell \geq 3$, in the forests with index q in L_α and L_β , such that their union $T_0 := T_1 \cup \dots \cup T_\ell$ contains a cycle in G_α and $T_0 \setminus T_i$ does not contain a cycle in G_α for all $i \in [\ell]$. It follows that $|V(T_i) \cap V(T_j)| \leq 1$ for all $i, j \in [\ell]$ with $i \neq j$. Let $M_1, \dots, M_\ell \in \mathcal{Y}_q^\beta \cup \mathcal{Y}_q^\gamma$,

such that segment M_i corresponds to tree T_i for all $i \in [\ell]$. Since T_0 contains a cycle in G_α , there exists an ordering π on the set $[\ell]$, such that $|V(T_{\pi(i)}) \cap V(T_{\pi(i+1)})| = 1$ for all $i = 1, \dots, \ell' - 1$ and $|V(T_{\pi(\ell)}) \cap V(T_{\pi(1)})| = 1$. Since $V(T_i) \cap V(T_j) \subseteq B$ for all $i, j \in [\ell]$ with $i \neq j$, it follows that $|M_{\pi(i)} \cap M_{\pi(i+1)}| = 1$ for all $i = 1, \dots, \ell' - 1$. Let v be the vertex such that $\{v\} = V(T_{\pi(1)}) \cap V(T_{\pi(\ell)})$. Since $V(T_{\pi(1)}) \cap V(T_{\pi(\ell)}) \subseteq B$, the segments $M_{\pi(1)}$ and $M_{\pi(\ell)}$ contain vertex v . Altogether, this contradicts condition (iv), and hence, there are no trees T_1, \dots, T_ℓ , $\ell \geq 3$, in the forests with index q in G_α and G_β such that their union $T_0 = T_1 \cup \dots \cup T_\ell$ contains a cycle in G_α .

We conclude that there are no two forests with index q in L_β and L_γ , such that their union contains a cycle, and thus, L_α is a partial solution for G_α . Moreover, by condition (ii), L_α induces signature \mathcal{X}^α . It follows that

$$\begin{aligned} \min_{(\mathcal{X}'^\beta, \mathcal{X}'^\gamma) \text{ compatible with } \mathcal{X}^\alpha} (T[\beta, \mathcal{X}'^\beta] + T[\gamma, \mathcal{X}'^\gamma]) &= T[\beta, \mathcal{X}^\beta] + T[\gamma, \mathcal{X}^\gamma] = c(L_\beta) + c(L_\gamma) \\ &= c(L_\alpha) \geq T[\alpha, \mathcal{X}^\alpha]. \end{aligned}$$

Running time. For each signature \mathcal{X}^α , we check all pairs of signatures $\mathcal{X}^\beta, \mathcal{X}^\gamma$ for node β and γ for compatibility, that means we check conditions (i)-(iv) for $O((|B_\beta| + 1)^{p \cdot |B_\beta|} \cdot (|B_\gamma| + 1)^{p \cdot |B_\gamma|})$ pairs of signatures with respect to the signature \mathcal{X}^α . Let $B := B_\alpha$. Recall that $B_\alpha = B_\beta = B_\gamma$.

For each pair, we can check condition (i) in $O(p \cdot |B|^3)$ time. We can check conditions (ii)-(iv) in $O(p \cdot |B|^3)$ time as follows.

For each $q \in [p]$, we construct a graph \hat{G}_q in the following way. We set $V(\hat{G}_q) := \{v_i \mid M_i \in \mathcal{Y}_q^\beta \cup \mathcal{Y}_q^\gamma\}$ and $E(\hat{G}_q) := \{\{v_i, v_j\} \in V(\hat{G}_q)^2 \mid |M_i \cap M_j| = 1, M_i, M_j \in \mathcal{Y}_q^\beta \cup \mathcal{Y}_q^\gamma\}$. We can construct the graph \hat{G}_q in $O(|B|^3)$ time. We can check condition (iii) while constructing graph \hat{G}_q . If condition (iv) does not hold, then there exists a cycle in \hat{G}_q . We can detect a cycle in \hat{G}_q in $O(|B|^2)$ time, for example by applying a depth-first search on \hat{G}_q , and thus, we can check condition (iv) in $O(|B|^2)$ time.

For condition (ii), we compare the corresponding segments of the vertex sets of the connected components in \hat{G}_q with the segments in \mathcal{Y}_q^α . Finding the connected components in \hat{G}_q can be done in $O(|B|^2)$ time, for example by applying a depth-first search in \hat{G}_q . The comparison of the segments can be done in $O(|B|^2)$ time. Thus, condition (ii) can be verified in $O(|B|^2)$ time. We conclude that for each $q \in [p]$, we can check conditions (ii)-(iv) in $O(|B|^3)$ time.

We can check conditions (i)-(iv) for each pair of signatures for node β and node γ in $O(p \cdot |B|^3)$ time. Therefore, the overall running time for filling all entries in T for a join node is in $O(p \cdot (\omega + 2)^{3 \cdot p \cdot (\omega + 1) + 3})$.

Now we describe how to fill the entries in the table T of the dynamic program according to each type of nodes in the tree decomposition \mathbb{T} .

Proof of Theorem 2. Let G be graph with $s, t \in V(G)$ given together with a tree decomposition $\mathbb{T}' = (T', (B'_\alpha)_{\alpha \in V(T')})$ of width $\omega' := \omega(\mathbb{T}')$ of G . We modify the tree decomposition \mathbb{T}' in polynomial time to a nice tree decomposition with introduce edge nodes of equal width, and add the vertices s and t to every bag. Let \mathbb{T} be the nice tree decomposition with introduce edge nodes and vertices s and t contained in every bag obtained from \mathbb{T}' . Note that $\omega := \omega(\mathbb{T}) \leq \omega' + 2$. We apply the dynamic program described above bottom-up on the tree decomposition \mathbb{T} . The dynamic program runs in $O(p \cdot (\omega + 2)^{3 \cdot p \cdot (\omega + 1) + 4} \cdot n)$ time. Since $\omega \leq \omega' + 2$, it follows that the dynamic program runs in $O(p \cdot (\omega' + 4)^{3 \cdot p \cdot (\omega' + 3) + 4} \cdot |V(G)|)$ time. Finally, we read out the minimum number of shared edges for p s - t routes in the entries of the root node in \mathbb{T} as follows.

Let τ be the root node of \mathbb{T} . Note that $\{s, t\} \subseteq B_\tau$. Let \mathcal{F} be the set of all signatures for node τ such that for all signatures $\mathcal{X}^\tau = (\mathcal{Y}_q^\tau, Z_q^\tau)_{q=1, \dots, p}$ in \mathcal{F} it holds that for all $q \in [p]$ there exists a segment $M \in \mathcal{Y}_q^\tau$ with $\{s, t\} \subseteq M$. Due to our construction, a segment of a segmentation corresponds to a tree in a partial solution for the given graph. Hence, a set of p segmentations, where for each of the p segmentations there exists a segment that contains the vertices s and t , corresponds to a solution for MINIMUM SHARED EDGES with p routes. Thus, the minimum number of shared edges for p s - t routes equals $\min_{\mathcal{X}^\tau \in \mathcal{F}} T[\tau, \mathcal{X}^\tau]$. \square

We remark that we can modify the dynamic program in such a way that we can solve the weighted variant of MINIMUM SHARED EDGES, that is, with weights $w : E(G) \rightarrow \mathbb{N}$ on the edge

set of the input graph. The cost of the partial solutions is the sum of the weights of shared edges, and thus the entry in the table of the dynamic program. For an introduce edge node, in the case of share-compatibility, we increase the value of the entry by the weight of the introduced edge. More precisely, for an introduce edge node α that introduces edge e and a signature \mathcal{X}^α for node α , the filling rule is adjusted by

$$T[\alpha, \mathcal{X}^\alpha] = \min \left(T[\beta, \mathcal{X}^\beta] + \begin{cases} w(e), & \text{if } \mathcal{X}^\beta \text{ and } \mathcal{X}^\alpha \text{ are share-compatible,} \\ 0, & \text{otherwise} \end{cases} \right),$$

where the minimum is taken over all signatures \mathcal{X}^β for node β compatible with \mathcal{X}^α .

5 Fixed-Parameter Tractability with Respect to the Number of Routes

In this section we prove the following.

Theorem 3. *MINIMUM SHARED EDGES is fixed-parameter tractable with respect to the number p of routes.*

The basic idea for the proof is to use treewidth reduction [21], a way to process a graph G containing terminals s, t in such a way that each minimal s - t separator of size at most $p - 1$ is preserved and the treewidth of the resulting graph is bounded by a function of p . The reason that this approach works is (we prove below) that each (p, s, t) -routing is characterized by its shared edges, and these are contained in minimal cuts of size at most $p - 1$. However, treewidth reduction preserves only minimal separators, that is, vertex sets, and not necessarily minimal cuts, that is, edge sets. Hence, we need to further process input graph and the graph coming out of the treewidth reduction process.

We now describe the approach in more detail; refer to Figure 7 for an overview of the following modifications and the graphs obtained in each step. Let (G, s, t, p, k) be an instance of MSE, where G is the input graph with $s, t \in V(G)$. First, we obtain a graph H by subdividing each edge in G . We denote by V_E the set of vertices obtained from the subdivisions. As a consequence, every minimal s - t cut in G of size at most $p - 1$ corresponds to a minimal s - t separator in H of size at most $p - 1$. Next, we apply the treewidth reduction technique to H , obtaining the graph H^* . By the treewidth reduction technique, graph H^* contains all minimal s - t separators in H of size at most $p - 1$ and the treewidth of graph H^* is upper-bounded by a function only depending on p . We denote by $V_E^* := \{v \in V(H^*) \mid v \in V_E\}$ the set of vertices in V_E which are preserved by the treewidth reduction technique in H^* . Finally, we contract an incident edge for each vertex in $V_E^* \subseteq V(H^*)$ to obtain the graph G^* . In the following, we modify step by step graph G to graph G^* . We discuss each step and we prove the properties of the obtained graphs described above. Finally, we give a proof of Theorem 3. We start with the following lemma which states that if our instance is a yes-instance, then we can find a solution where each of the shared edges is part of a minimal s - t cut of size smaller than the number p of routes.

Lemma 4. *If (G, s, t, p, k) is a yes-instance of MSE and G has a minimal s - t cut of size smaller than p , then there exists a solution $F \subseteq E$ such that each $e \in F$ is in a minimal s - t cut of size smaller than p in G .*

Recall that if G does not have a minimal s - t cut of size smaller than p , then we can find p s - t routes without sharing an edge. In the following proof, we make use of the following equivalent formulation of MSE based on edge contractions. Given an undirected graph $G = (V, E)$, $s, t \in V(G)$, $p \in \mathbb{N}$, and $k \in \mathbb{N}_0$, the question is whether there is a subset $F \subseteq E$ of edges of cardinality at most k in G such that the graph G/F with unit edge capacities allows an s - t flow of value at least p . In the following, we call such a set F a *solution*. Using Menger's theorem, one can obtain with small effort the equivalence of MSE and the problem above.

Proof of Theorem 4. We make use of the contraction equivalent of MSE. We show that for every minimal solution for MSE it holds that each edge of the solution is part of a minimal s - t cut of size smaller than p , where a solution is minimal if it is not a superset of another solution.

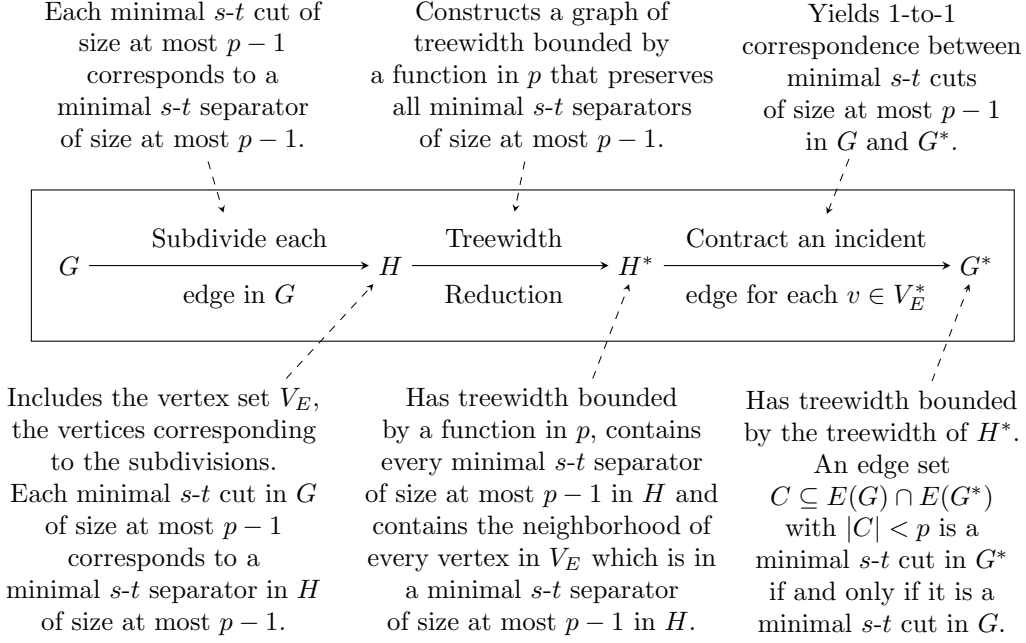


Figure 7: Overview of the strategy behind the proof of Theorem 3.

Let $G = (V, E)$ be the graph. Let (G, s, t, p, k) be a yes-instance of MSE. Then there exists a solution $L \subseteq E$, $|L| \leq k$, such that graph $G_L := G/L$ with unit edge capacities allows a maximum s - t flow of value at least p . We call a solution L minimal if there is no edge $e \in L$ such that graph $G/(L \setminus \{e\})$ with unit edge capacities allows a maximum s - t flow of value at least p .

Let L be a minimal solution and let $e \in L$. Suppose that e is not part of a minimal s - t cut of size smaller than p in G . Let $L' := L \setminus \{e\}$ and $G_{L'} := G/L'$. We consider the following two cases.

Case 1: The maximum s - t flow of $G_{L'}$ has value smaller than p . Then, using the max-flow min-cut theorem, $G_{L'}$ has an s - t cut C of size smaller than p . Since $e \notin C$, contracting edge e in $G_{L'}$ does not affect cut C . Therefore, C is also an s - t cut of size smaller than p in G_L and, again by the max-flow min-cut theorem, this implies a maximum s - t flow of value smaller than p in G_L . This is a contradiction to the fact that L is a solution.

Case 2: The maximum s - t flow of $G_{L'}$ has value at least p . Then L' is a solution, which contradicts the minimality of L .

Since $|L| \leq k$ and each edge in L is in a minimal s - t cut of size smaller than p in G , this completes the proof. \square

As mentioned before, as part of our approach we use the treewidth reduction technique [21]. Given a graph $G = (V, E)$ with $T = \{s, t\} \subseteq V(G)$ and an integer $\ell \in \mathbb{N}$, first the treewidth reduction technique computes the set C of vertices containing all vertices in G which are part of a minimal s - t separator of size at most ℓ in G . Then, it constructs the so-called *torso* of graph G given C and T , that is, the induced subgraph $G[C \cup T]$ with additional edges between each pair of vertices $v, w \in C \cup T$ with $\{v, w\} \notin E(G)$ if there is a v - w path in G whose internal vertices are not contained in $C \cup T$. Finally, each of these additional edges is subdivided and ℓ additional copies of each of that subdivisions are introduced, that is, if $\{v, w\}$ is one of these additional edges, then the vertices $x_1^{vw}, \dots, x_{\ell+1}^{vw}$ are added and edge $\{v, w\}$ is replaced by the edges $\{v, x_1^{vw}\}, \dots, \{v, x_{\ell+1}^{vw}\}, \{x_1^{vw}, w\}, \dots, \{x_{\ell+1}^{vw}, w\}$. In the following, we denote these paths by *copy paths*. The resulting graph contains all minimal s - t separators of size at most ℓ in G and has treewidth upper-bounded by $h(\ell)$ for some function h only depending on ℓ .

Theorem 5 (Treewidth reduction [21, Theorem 2.15]). *Let G be a graph, $T \subseteq V(G)$, and let ℓ be an integer. Let C be the set of all vertices of G participating in a minimal s - t separator of size at most ℓ for some $s, t \in T$. For every fixed ℓ and $|T|$, there is a linear-time algorithm that computes a graph G^* having the following properties:*

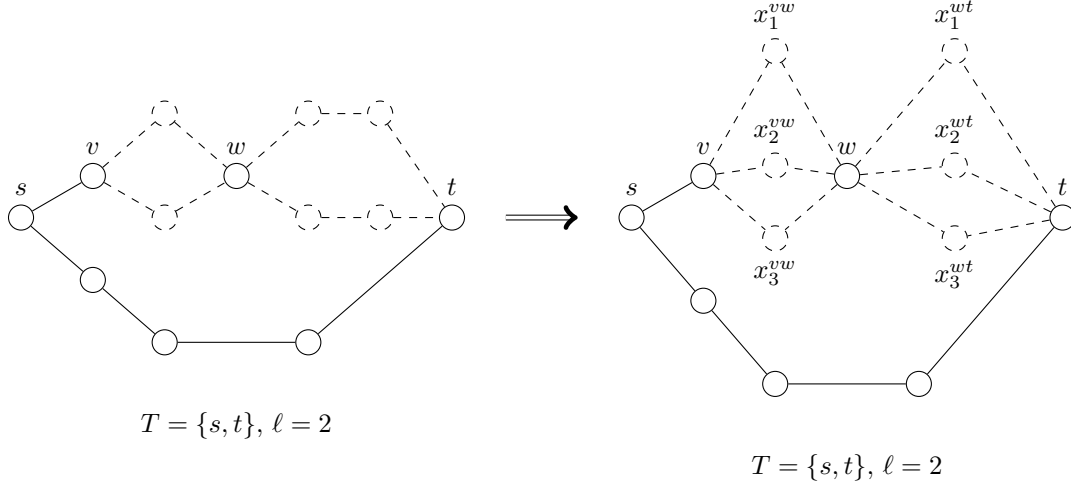


Figure 8: Example for the treewidth reduction technique.

- (1) $C \cup T \subseteq V(G^*)$.
- (2) For every $s, t \in T$, a set $L \subseteq V(G^*)$ with $|L| \leq \ell$ is a minimal s - t separator of G^* if and only if $L \subseteq C \cup T$ and L is a minimal s - t separator of G .
- (3) The treewidth of G^* is at most $h(\ell, |T|)$ for some function h .
- (4) $G^*[C \cup T]$ is isomorphic to $G[C \cup T]$.

Figure 8 shows an example for the application of the treewidth reduction technique. We use dashed edges and vertices to highlight the changes when applying the treewidth reduction technique with $T = \{s, t\}$ and parameter $\ell = 2$. On the left-hand side, the original graph is shown. On the right-hand side, the resulting graph after applying the treewidth reduction technique with $T = \{s, t\}$ and $\ell = 2$ on the left-hand side graph is shown.

For finding a p -routing we are interested in minimal s - t cuts of size smaller than p in G . The treewidth reduction technique guarantees to preserve minimal s - t separators of a specific size, but does not guarantee to preserve minimal s - t cuts of a specific size. Thus, we need to modify our graph G in such a way that each minimal s - t cut in G corresponds to a minimal s - t separator in the modified graph. We modify graph G in the following way.

Step 1. Subdivide each edge in $E(G)$, that is, for each edge $e = \{v, w\}$ in $E(G)$ add a vertex x_e and replace edge e by edge $\{v, x_e\}$ and edge $\{x_e, w\}$. We say that vertex x_e as well as edges $\{v, x_e\}$ and $\{x_e, w\}$ correspond to edge e . Let $V_E := \{x_e \mid e \in E\}$ and E' be the edge set replacing the edges in E . Then $H := (V \cup V_E, E')$ is the resulting graph.

Note that each edge in H is incident with exactly one vertex in V_E and one vertex in V . Thus, no two vertices in V_E and no two vertices in V are neighbors. Moreover, note that each vertex in V_E has degree exactly two. It holds that $|V \cup V_E| = |V| + |E|$ and $|E'| = 2 \cdot |E|$.

Lemma 6. (G, s, t, p, k) is a yes-instance of MSE if and only if $(H, s, t, p, 2k)$ is a yes-instance of MSE.

Proof. Intuitively, every edge in G corresponds to two edges in H and every two edges in H both incident with an vertex in V_E correspond to an edge in G .

“ \Rightarrow ”: Consider a solution for the yes-instance (G, s, t, p, k) of MSE. For each edge $e = \{v, w\} \in E(G)$ that is shared in the solution, consider the corresponding two edges $\{v, x_e\}$ and $\{x_e, w\}$ in graph H . Sharing these at most $2k$ edges yields a solution for instance $(H, s, t, p, 2k)$ of MSE.

“ \Leftarrow ”: Consider a minimal solution for the yes-instance $(H, s, t, p, 2k)$. Observe that in such a solution, a vertex in V_E is incident with either no or two shared edges. Each vertex in V_E that appears in at least two s - t routes is incident with two shared edges. Each vertex in V_E corresponds to one edge in G . Let $F \subseteq E(G)$ be the set of edges such that $e = \{v, w\} \in F$ if the edges $\{v, x_e\}$ and $\{x_e, w\}$ in $E(H)$ are shared in the solution for $(H, s, t, p, 2k)$. Note that $|F| \leq k$ since there are at most $2k$ shared edges. Thus, F is a solution for instance (G, s, t, p, k) of MSE. \square

Recall that we are interested in s - t cuts in G . By our modification from Step 1 of G to H , for each edge in G there is a corresponding vertex in V_E in H . The following lemma gives a one-to-one correspondence between s - t cuts in G and those s - t separators in H that contain only vertices in V_E .

Lemma 7. *If C is an s - t cut in G , then $V_C := \{w \in V_E \mid w \text{ corresponds to } e \in C\}$ is an s - t separator in H . If $W \subseteq V_E$ is an s - t separator in H , then $C_W := \{e \in E \mid e \text{ corresponds to } w \in W\}$ is an s - t cut in G .*

Proof. Let C be an s - t cut in G . Suppose that the set $V_C := \{w \in V_E \mid w \text{ corresponds to } e \in C\}$ is not an s - t separator in H . Then there exists a path P' avoiding V_C in H connecting s and t . Since no two vertices in V_E are neighbors and no two vertices in V are neighbors, the vertices in path P' alternate in V and V_E . Since we know that the vertices in V_E correspond to edges in G , $P := P' \cap V$ describes a path in G connecting s and t avoiding all edges in C . This is a contradiction to the fact that C is an s - t cut in G , and hence set V_C is an s - t separator in H .

Let $W \subseteq V_E$ be an s - t separator in H . Suppose that the set $C_W := \{e \in E \mid e \text{ corresponds to } w \in W\}$ is not an s - t cut in G . Then there exists a path P avoiding C_W in G connecting s and t . Let $V_P \subseteq V(H)$ be the set of vertices in H such that each vertex in V_P either corresponds to an edge in P or is an endpoint of an edge in P . We remark that $W \cap V_P = \emptyset$. Moreover, set V_P is the set of vertices of an s - t route in H . This is a contradiction to the fact that W is an s - t separator in H , and hence set C_W is an s - t cut in G . \square

In the following lemma, we show that Theorem 7 holds also for minimal s - t cuts and minimal s - t separators. This is important, since we will use a combination of the treewidth reduction technique and Theorem 4 later on.

Lemma 8. *Every minimal s - t cut in G corresponds to a minimal s - t separator in H .*

Proof. Let C be a minimal s - t cut in G . By Theorem 7, we know that $V_C := \{w \in V_E \mid w \text{ corresponds to } e \in C\}$ is an s - t separator in H . If V_C is a minimal s - t separator in H , then we are done. Thus, suppose that V_C is an s - t separator in H , but V_C is not a minimal s - t separator in H . Then there exists a vertex $w \in V_C$ such that $V_C \setminus \{w\}$ is an s - t separator in H . Let $e \in C$ be the edge in G corresponding to vertex w . Since $V_C \setminus \{w\} \subseteq V_E$, again by Theorem 7 we know that $C \setminus \{e\}$ is an s - t cut in G . This is a contradiction to the fact that C is a minimal s - t cut in G , and hence, V_C is a minimal s - t separator in H . \square

We know that each minimal s - t cut in G corresponds to a minimal s - t separator in H . Next, we show that every vertex in the neighborhood of each minimal s - t separator containing only vertices in V_E belongs to a minimal s - t separator. Recall that for $W \subseteq V$ we denote by $N_G(W)$ the open neighborhood of the vertex set W in G and by $N_G[W] := W \cup N_G(W)$ the closed neighborhood of the vertex set W in G .

Lemma 9. *Let $W \subseteq V_E \subseteq V(H)$ be the set of vertices corresponding to a minimal s - t cut of size at most $\ell \in \mathbb{N}$ in G . Then, each vertex in $N_H[W]$ is part of a minimal s - t separator of size at most ℓ in H .*

Proof. Let $W \subseteq V_E \subseteq V(H)$ be given such that W corresponds to a minimal s - t cut in G of size at most ℓ . Note that by Theorem 8, W is a minimal s - t separator in H . Let x be an arbitrary vertex in $N_H(W)$. First, we show that $W' := (W \setminus N_H(x)) \cup \{x\}$ is an s - t separator in H .

Suppose that W' is not an s - t separator in H . Then there exists an s - t path P in $H - W'$. Note that each vertex in $W \cap N_H(x)$ is incident with vertex x and exactly one other vertex in $V(H)$. Thus, no vertex in $W \cap N_H(x)$ appears in path P . Hence, P is an s - t path in $H - W$. This is a contradiction to the fact that W is an s - t separator in H , and hence, W' is an s - t separator in H .

Next, we show that if W' is not a minimal s - t separator in H , then there exists a set $U \subseteq W' \setminus \{x\}$ such that $W' \setminus U$ is a minimal s - t separator in H . Let W' be an s - t separator in H , but not a minimal s - t separator in H . Suppose that for all $U \subseteq W' \setminus \{x\}$ it holds that $W' \setminus U$ is not a minimal s - t separator. Then there exists a set $X \subseteq W'$ with $x \in X$ such that $W' \setminus X$ is a minimal s - t separator in H . Since $W' \setminus X = W \setminus (N_H(x) \cap W) \setminus X \subseteq W$, this contradicts the fact

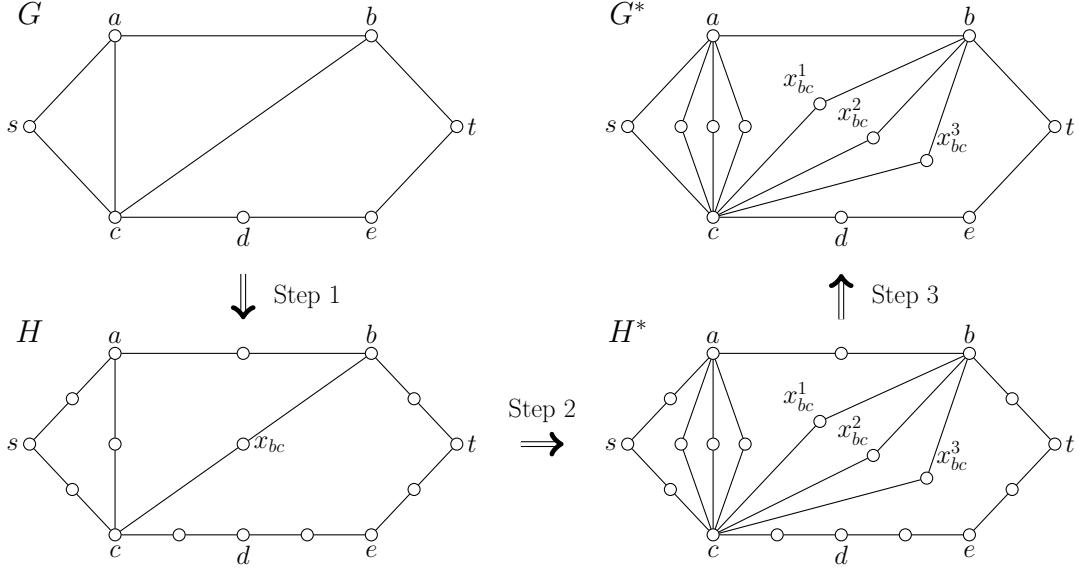


Figure 9: Example of Steps 1 to 3 on the example graph G (top-left) with $T = \{s, t\}$ and $p = 3$.

that W is a minimal s - t separator in H . Hence, there exists a set $U \subseteq W' \setminus \{x\}$ such that $W' \setminus U$ is a minimal s - t separator in H .

Let $U \subseteq W' \setminus \{x\}$ be a set such that $W'' := W' \setminus U$ is a minimal s - t separator. Since $x \in W''$ and $|W''| \leq |W'| \leq |W|$, vertex x appears in a minimal s - t separator in H of size at most ℓ . Since vertex x was chosen arbitrarily in $N_H(W)$, each vertex in $N_H[W]$ is part of a minimal s - t separator of size at most ℓ in H . \square

We obtained graph H from graph G by applying Step 1. By Theorem 8, we know that each minimal s - t cut in G corresponds to a minimal s - t separator in H . Moreover, by Theorem 9, if we consider a minimal s - t cut of size smaller than p in G , then, for each neighbor of the vertex set in H corresponding to the minimal s - t cut in G , there exists a minimal s - t separator of size smaller than p in H that contains that neighbor. As the next step (cf. Figure 7) we apply the treewidth reduction technique [21] on graph H .

Step 2. Apply the treewidth reduction (Theorem 5) to graph H with $T = \{s, t\}$ and $p - 1$ as upper bound for the size of the minimal s - t separators. Denote the resulting graph by H^* .

Let $V_E^* := \{v \in V(H^*) \mid v \in V_E\}$. Graph H^* contains all minimal s - t separators of size at most $p - 1$ in H . By Theorem 8, every minimal s - t cut of size at most $p - 1$ in G corresponds to a minimal s - t separator of size at most $p - 1$ in H and thus, by Theorem 5, to a minimal s - t separator of size at most $p - 1$ in H^* . By Theorem 9, the neighborhood of each vertex in H corresponding to a vertex in V_E^* is contained in the vertex set $V(H^*)$. As a consequence, we can reconstruct each edge in graph G that appears in a minimal s - t cut of size at most $p - 1$ in G as an edge in the graph H^* . As our next step (cf. Figure 7), we contract for each vertex in V_E^* an incident edge in graph H^* . We remark that if x^{vw} is a vertex in V_E^* , then the only edges incident with vertex x^{vw} are $\{v, x^{vw}\}$ and $\{x^{vw}, w\}$. In addition, the vertices v and w are the only neighbors of x^{vw} in graph H and in graph H^* .

Step 3. Contract for each vertex in V_E^* exactly one incident edge in H^* to obtain the graph G^* . In other words, undo the subdivision applied on G to obtain H .

We remark that $\text{tw}(G^*) \leq \text{tw}(H^*)$, since edge contraction does not increase the treewidth of a graph [26].

In Figure 9, we illustrate Steps 1 to 3 on an example graph G with $T = \{s, t\}$ and $p = 3$. The top-left graph is the original graph G . The bottom-left graph is graph H , obtained from G by applying Step 1. The bottom-right graph is graph H^* , obtained from H by applying Step 2. The top-right graph is the final graph G^* , obtained from H^* by applying Step 3.

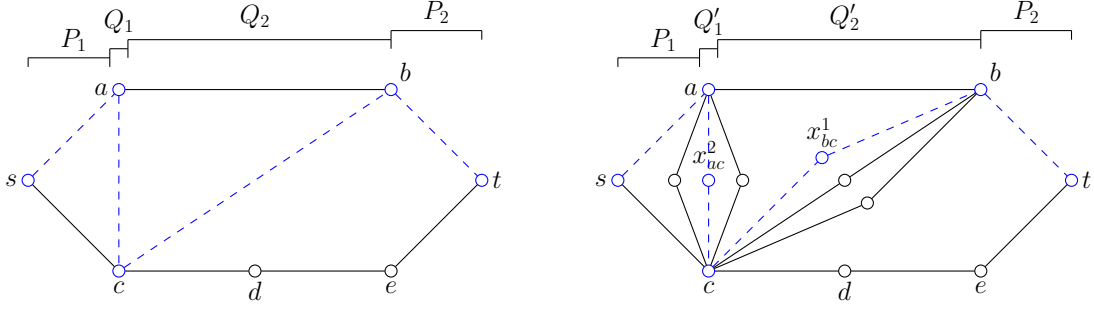


Figure 10: The graphs G (left-hand side) and G^* (right-hand side) from Figure 9. The dashed edges belong to an s - t path in G and G^* respectively. The upper braces show the range of the consecutive subpaths P_1, Q_1, Q_2, P_2 for G and P_1, Q'_1, Q'_2, P_2 for G^* .

Let $e = \{v, w\} \in E(G)$ be an edge in G and $x_e \in V_E \subseteq V(H)$ be the corresponding vertex in H . Then $\{v, x_e\}$ and $\{x_e, w\}$ are the incident edges of x_e in H . If $x_e \in V(H^*)$, then one of the incident edges $\{v, x_e\}$ and $\{x_e, w\}$ with vertex x_e is contracted and yields edge $\{v, w\} \in E(G^*)$. We say that the edges $\{v, w\} \in E(G)$ and $\{v, w\} \in E(G^*)$ correspond one-to-one, and, for example, we write $\{v, w\} \in E(G) \cap E(G^*)$.

Considering the graphs G and G^* , we show that, given an s - t path in the one graph, we can find an s - t path in the other graph using a common set of edges in $E(G) \cap E(G^*)$.

Lemma 10. (i) If P is an s - t path in G , then there exists an s - t path P^* in G^* that contains all edges in $E(P) \cap E(G^*)$.

(ii) If P^* is an s - t path in G^* , then there exists an s - t path P in G that contains all edges in $E(P^*) \cap E(G)$.

Proof. (i): Let P be an s - t path in G . If P just contains edges in $E(G) \cap E(G^*)$, then we set $P^* = P$. If P contains edges in $E(G) \setminus E(G^*)$, then P has a representation of consecutive subpaths P_i , $1 \leq i \leq j$, and Q_i , $1 \leq i \leq \ell$, where $\{P_i\}_{1 \leq i \leq j}$ is the set of subpaths of P that just contain edges in $E(G) \cap E(G^*)$ and $\{Q_i\}_{1 \leq i \leq \ell}$ is the set of subpaths of P with endpoints in $V(G) \cap V(G^*)$, inner vertices in $V(G) \setminus V(G^*)$ and edges in $E(G) \setminus E(G^*)$. Since for each $1 \leq i \leq \ell$, path Q_i is connecting two vertices $v, w \in V(G) \cap V(G^*)$ in G , there are p edge-disjoint paths of length 2 in G^* connecting v and w using the edges in $E(G^*) \setminus E(G)$, that are the copy paths. For each $i \in [\ell]$, let Q'_i be one of the copy paths connecting the endpoints of Q_i . Figure 10 illustrates this correspondence on an example graph. Replacing each Q_i by such a path Q'_i in G^* yields a path P^* with consecutive subpaths P_i , $1 \leq i \leq j$, and Q'_i , $1 \leq i \leq \ell$, in G^* connecting s and t that contains all edges in $E(P) \cap E(G^*)$.

(ii): Let P^* be an s - t path in G^* . If P^* just contains edges in $E(G) \cap E(G^*)$, then we set $P = P^*$. If P^* contains edges in $E(G^*) \setminus E(G)$, then P^* has a representation of consecutive subpaths P'_i , $1 \leq i \leq j$, and Q'_i , $1 \leq i \leq \ell$, where $\{P'_i\}_{1 \leq i \leq j}$ is the set of subpaths of P^* that just contain edges in $E(G^*) \cap E(G)$ and $\{Q'_i\}_{1 \leq i \leq \ell}$ is the set of subpaths of P^* with endpoints in $V(G^*) \cap V(G)$, inner vertices in $V(G^*) \setminus V(G)$, and edges in $E(G^*) \setminus E(G)$. We remark that each Q'_i is one of the copy paths in G^* . By construction of G^* , each Q'_i connects two vertices in $V(G^*) \cap V(G)$ that are connected by a path in G with no inner vertices in $V(G^*) \cap V(G)$. Therefore, for each $i \in [\ell]$, we can replace path Q'_i by such a path Q_i in G . This yields an s - t path P in G with consecutive subpaths P'_i , $1 \leq i \leq j$, and Q_i , $1 \leq i \leq \ell$, that contains all edges in $E(P^*) \cap E(G)$. \square

We modified graph G to graph G^* by applying Steps 1 to 3. By Theorem 10, we can construct s - t routes in G and G^* that use edges in the common set of edges $E(G) \cap E(G^*)$. The next lemma states that each minimal s - t cut of size smaller than p in one of the graphs G and G^* is also a minimal s - t cut of size smaller than p in the other graph.

Lemma 11. Let $C \subseteq E(G) \cap E(G^*)$. Edge set C is a minimal s - t cut in G of size smaller than p if and only if C is a minimal s - t cut in G^* of size smaller than p .

Proof. We make use of Theorem 10 in the following proof. We remark that no edge in $E(G^*) \setminus E(G)$ is in any minimal s - t cut of size smaller than p in G^* since, by the treewidth reduction technique, for each of these edges there are $p - 1$ copies in G^* .

“ \Rightarrow ”: Let C be a minimal s - t cut of size smaller than p in G . By Theorem 8, C has a corresponding minimal s - t separator S_C of size smaller than p in H . By the treewidth reduction technique, S_C is a minimal s - t separator in H^* . By Theorem 9, every neighbor of S_C is contained in H^* . By our contraction of edges of H^* to G^* , for each vertex of S_C an incident edge is contracted and yields the edge set C again. Since S_C is a minimal s - t separator in H^* of size smaller than p and each vertex in S_C has degree exactly two, set C is a minimal s - t cut in G^* of size smaller than p .

“ \Leftarrow ”: Let C be a minimal s - t cut in G^* of size smaller than p . Suppose C is not a minimal s - t cut in G of size smaller than p . We distinguish two cases.

Case 1: C is not an s - t cut in G . Then there exists a path P in G connecting s and t avoiding the edges in C . By Theorem 10, there exists an s - t path P^* in G^* that contains all edges in $E(P) \cap E(G^*)$. Since no edge in $E(G^*) \setminus E(G)$ is in any minimal s - t cut of size at most $p - 1$ of G^* , P^* avoids the edges in C . This is a contradiction to the fact that C is a minimal s - t cut in G^* .

Case 2: C is an s - t cut in G , but C is not a minimal s - t cut in G . Then there exists $e \in C$ such that $C' := C \setminus \{e\}$ is an s - t cut in G . Since C is a minimal s - t cut in G^* , the set C' is not an s - t cut in G^* . Thus, there exists an s - t path P^* in G^* that avoids the edges in C' . By Theorem 10, there exists an s - t path P in G that contains all the edges in $E(P^*) \cap E(G)$. Since no edge in $E(G) \setminus E(G^*)$ is in any minimal s - t cut of size at most $p - 1$ in G , path P avoids the edges in C' . Therefore, set C' is not an s - t cut in G , and thus, C is a minimal s - t cut in G . \square

Recalling Theorem 4, we know that if an instance of MSE is a yes-instance, then we can find k edges such that the k edges form a solution for the instance and each of the k edges is part of a minimal s - t cut of size smaller than p in G . By Theorem 11, the graphs G and G^* have the same set of minimal s - t cuts of size smaller than p in common. Combining Theorem 4 and Theorem 11 leads to the following lemma.

Lemma 12. (G^*, s, t, p, k) is a yes-instance of MSE if and only if (G, s, t, p, k) is a yes-instance of MSE.

Proof. We make use of the contraction equivalent of MSE.

“ \Rightarrow ”: Let (G^*, s, t, p, k) be a yes-instance of MSE. By Theorem 4, we find a solution $F \subseteq E(G^*)$ such that each edge in F is part of a minimal s - t cut in G^* of size smaller than p . It follows that $F \subseteq E(G) \cap E(G^*)$, since by our construction no edge in $(E(G^*) \setminus E(G))$ is part of a minimal s - t cut of size smaller than p in G^* . Let $G_F := G/F$ be the graph G with all edges in F contracted. Suppose that G_F with unit edge capacities allows a maximum s - t flow of value smaller than p . Then there exists a minimal s - t cut C of size smaller than p in G_F . By Theorem 11, C is also a minimal s - t cut of size smaller than p in $G_F^* := G^*/F$. This is a contradiction to the fact that the value of any maximum s - t flow in G_F^* with unit edge capacities is at least p , and hence, set F is a solution for instance (G, s, t, p, k) .

“ \Leftarrow ”: Let (G, s, t, p, k) be a yes-instance of MSE. By Theorem 4, we find a solution $F \subseteq E(G)$ such that each edge in F is part of a minimal s - t cut in G of size smaller than p . It follows that $F \subseteq E(G) \cap E(G^*)$. Suppose that $G_F^* := G^*/F$ with unit edge capacities allows a maximum s - t flow of value smaller than p . Then there exists a minimal s - t cut C of size smaller than p in G_F^* . By Theorem 11, C is a minimal s - t cut of size smaller than p in $G_F := G/F$. This is a contradiction to the fact that the value of any maximum s - t flow in G_F with unit edge capacities is at least p , and hence, set F is a solution for instance (G^*, s, t, p, k) . \square

By Theorem 12, we know that the instances (G^*, s, t, p, k) and (G, s, t, p, k) are equivalent for MSE. By our construction, we know that the treewidth of G^* is upper-bounded by a function only depending on the number p of routes. In addition, we know that MINIMUM SHARED EDGES is fixed-parameter tractable with respect to the number p of routes and an upper bound on the treewidth of the input graph. Thus, we are ready to prove Theorem 3.

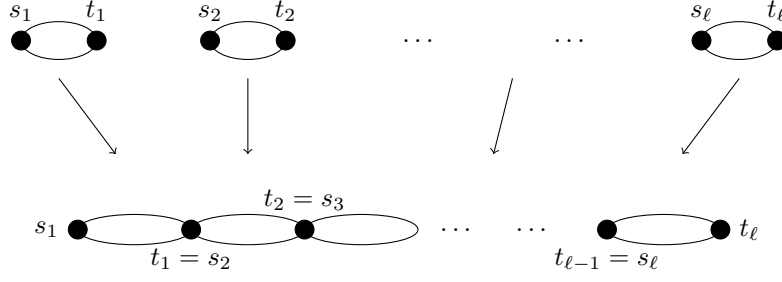


Figure 11: OR-cross-composition of ℓ instances of AMSE into one instance of MSE.

Proof of Theorem 3. First we modify our graph $G = (V, E)$ by applying Steps 1 to 3. Let H , H^* , and G^* be the according graphs. By Theorem 5, the treewidth of H^* is upper-bounded by $h(p)$ for some function h . Since edge contractions do not increase the treewidth of a graph [26], it follows that $\text{tw}(G^*) \leq \text{tw}(H^*)$. By Theorem 12, the instances (G^*, s, t, p, k) and (G, s, t, p, k) are equivalent for MSE.

We know from Theorem 2 that $\text{MSE}(p, \omega)$ is fixed-parameter tractable when parameterized by the number p of routes and by an upper bound ω on the treewidth of the input graph. Since function h only depends on p and $h(p)$ is upper-bounding the treewidth of graph G^* , we can solve instance (G^*, s, t, p, k) in $f(p) \cdot O(|V(G^*)|)$ time, where f is a computable function only depending on parameter p . Since $|V(G^*)| \leq |V(G)| + p \cdot |E(G)| \leq p \cdot |G|$ and the instances (G^*, s, t, p, k) and (G, s, t, p, k) are equivalent for MSE, we can decide instance (G, s, t, p, k) in $f(p) \cdot p \cdot O(|G|)$ time, that is, in FPT-time. \square

Using the dynamic program from Section 4 the running time of the above algorithm amounts to $O(p^2 \cdot (h(p) + 4)^{3 \cdot p \cdot (h(p) + 3) + 3} \cdot |G|)$. Using the bound $h(p) \leq 2^{O(p^2)}$ [21], we obtain a running time of $2^{p^3 \cdot 2^{O(p^2)}} \cdot (n + m)$.

6 No Polynomial Problem Kernel for the Parameter Number of Routes

In the previous section, we showed that MINIMUM SHARED EDGES is fixed-parameter tractable with respect to the number p of routes. It is well known that a problem is fixed-parameter tractable if and only if it admits a problem kernel. Of particular interest is the minimal possible size of a problem kernel. Accordingly, in this section we prove the following lower bound.

Theorem 13. MINIMUM SHARED EDGES does not admit a polynomial-size problem kernel with respect to the number p of routes, unless $\text{NP} \subseteq \text{coNP}/\text{poly}$.

We prove Theorem 13 via an *OR-cross-composition* [4], that is, given ℓ instances of an NP-hard problem Q , all contained in one equivalence class of a polynomial-time computable relation \mathcal{R} of our choosing, we compute in polynomial-time an instance (G, s, t, p, k) of MSE such that (i) p is bounded by a polynomial function of the size of the largest input instance plus $\log(\ell)$ (*boundedness*), and (ii) (G, s, t, p, k) is a yes-instance if and only if one of the input instances is a yes-instance (*correctness*). If this is possible, then MSE does not admit a polynomial-size problem kernel with respect to p unless $\text{NP} \subseteq \text{coNP}/\text{poly}$ [4].

It is tempting to use MSE itself as the problem Q , to assume that each of the instances asks for the same number of routes and same number of shared edges by virtue of \mathcal{R} , and to OR-cross-compose by simply gluing the graphs in a chain-like fashion on sinks and sources (see Fig. 11). This fulfills the boundedness constraint, but not necessarily the correctness constraint, since the instances can share shared edges between them. That is, the shared edges in any instance can be as large as the number of edges of the graph in the instance. Hence, we use the following problem as the problem Q instead.

ALMOST MINIMUM SHARED EDGES (AMSE)

Input: An undirected graph G , two distinct vertices $s, t \in V(G)$, and two integers $p, k \in \mathbb{N}$ such that G has a (p, s, t) -routing with at most $k + 1$ shared edges.

Question: Is there a (p, s, t) -routing in G with at most k shared edges?

Proposition 14. ALMOST MINIMUM SHARED EDGES is NP-hard.

Theorem 14 can be proven via a reduction from MSE to AMSE that introduces an additional path of length $k + 1$ connecting s and t . As a technical remark, since any instance of MSE with $k = 0$ is solvable in polynomial time, we assume here and in the following that $k > 0$.

Proof of Theorem 14. Let (G, s, t, p, k) be an instance of MSE. We describe the construction of an instance of AMSE given (G, s, t, p, k) and prove their equivalence.

Construction. We add a P_{k+2} with endpoints s and t to G to obtain the graph G' . Note that we can route any number ≥ 2 of routes over the P_{k+2} from s to t while sharing exactly $(k + 1)$ edges. Thus, $(G', s, t, p + 1, k + 1)$ is a yes-instance of MSE, and $(G', s, t, p + 1, k)$ is an instance of AMSE. We show that (G, s, t, p, k) is a yes-instance of MSE if and only if $(G', s, t, p + 1, k)$ is a yes-instance of AMSE.

Correctness. Let (G, s, t, p, k) be a yes-instance of MSE. Let \mathcal{P} be a (p, s, t) -routing in G sharing at most k edges. Since $G'[V(G)] = G$, \mathcal{P} is also a (p, s, t) -routing in G' sharing at most k edges. Since the additional P_{k+2} in G' is not contained in an s - t route in \mathcal{P} , we can construct an additional s - t route P in G' using only this $(k + 1)$ -chain without sharing an additional edge. Thus, $\mathcal{P} \cup \{P\}$ is a $(p + 1, s, t)$ -routing in G' sharing at most k edges. That is, $(G', s, t, p + 1, k)$ is a yes-instance of AMSE.

Conversely, let $(G', s, t, p + 1, k)$ be a yes-instance of AMSE. Observe that at most one s - t route appears on the $(k + 1)$ -chain in G' . Thus, at least p s - t routes share at most k edges in $G'[V(G)] = G$. It follows that (G, s, t, p, k) is a yes-instance of MSE. \square

If we OR-cross-compose ℓ instances of AMSE instead, we know that if the resulting instance has a routing with $\ell(k + 1) - 1$ shared edges, then without loss of generality each of the original instances contributes at most $k + 1$ shared edges. This means that at least one of the original instances is a yes-instance, giving the correctness of the OR-cross-composition.

Proof of Theorem 13. We describe an OR-cross-composition as sketched in Figure 11. We cross-compose ℓ instances of AMSE into one instance of MSE(p). We define the relation \mathcal{R} as follows: $(G, s, t, p, k) \equiv_{\mathcal{R}} (G', s', t', p', k')$ if $p = p'$, and $k = k'$. Obviously, to check whether two instances are equivalent with respect to \mathcal{R} can be done in constant time. Moreover, in any finite set of instances, the number of equivalence classes with respect to \mathcal{R} is upper-bounded by the product of the largest p value and largest k value over all instances in the finite set. Thus, \mathcal{R} is a polynomial equivalence relation. We cross-compose ℓ \mathcal{R} -equivalent instances $(G_i, s_i, t_i, p, k)_{i=1, \dots, \ell}$ of AMSE to an instance of MSE(p) as follows. In the following, let $I_j := (G_j, s_j, t_j, p, k)$ for all $j \in [\ell]$.

Construction. We join the graphs G_1, \dots, G_ℓ in a chain-like fashion, that is, we identify t_{i-1} and s_i for each $i = 2, \dots, \ell$. Let G^* be the obtained graph. Let $I^* := (G^*, s_1, t_\ell, p^*, k^*)$ with $p^* = p$ and $k^* = \ell \cdot (k + 1) - 1$ be the instance of MSE. Recall that p^* is the parameter. We claim that I^* is a yes-instance of MSE if and only if at least one instance I_i is a yes-instance of AMSE.

Correctness. Let I^* be a yes-instance of MSE. Observe that each $t_{i-1} = s_i$ for $i = 2, \dots, \ell$ is a 1-separator in G^* . Let G_i^* be the graph induced by all vertices that are “between” s_i and t_i : these are all vertices that are reachable in G from s_i and t_i without touching the other. Let $S_i \subseteq G^* - \{t_i\}$ with $s_i \in V(S_i)$, that is S_i is the connected component of $G - \{t_i\}$ that contains s_i . Analogously, let $T_i \subseteq G^* - \{s_i\}$ with $t_i \in V(T_i)$. We define $G_i^* := G^*[\{s_i, t_i\} \cup (V(S_i) \cap V(T_i))]$. Observe that G_i^* is isomorphic to G_i for all $i = 1, \dots, \ell$, and G_i^* and G_j^* are edge-disjoint for all $i, j \in [\ell]$ with $i \neq j$. Moreover, $V(G^*) = \bigcup_{1 \leq i \leq \ell} V(G_i^*)$ and $E(G^*) = \bigcup_{1 \leq i \leq \ell} E(G_i^*)$.

Since I^* is a yes-instance of MSE, there are at most $\ell \cdot (k + 1) - 1$ shared edges in any solution to I^* . Suppose one can find at least $(k + 1)$ shared edges in each G_i^* . Since G_i^* and G_j^* are edge-disjoint for all $i, j \in [\ell]$ with $i \neq j$, it follows that there are at least $\ell \cdot (k + 1) > \ell \cdot (k + 1) - 1 = k^*$ shared edges, contradicting the fact that I^* is a yes-instance. Thus, there exists an index $j \in [\ell]$ such that there are at most k shared edges in G_j^* , or equivalently, I_j is a yes-instance of AMSE.

Conversely, let $j \in [\ell]$ such that I_j is a yes-instance of AMSE. By construction, sharing exactly the same k edges of a solution to I_j allows p s_j - t_j routes in G_j^* . We know that for each $i \in [\ell] \setminus \{j\}$, instance I_i of AMSE allows p s_i - t_i routes sharing at most $k + 1$ edges. Thus, we can route p s_1 - t_ℓ routes through G^* sharing at most

$$k + (\ell - 1) \cdot (k + 1) = \ell \cdot k + \ell - 1 = \ell \cdot (k + 1) - 1 = k^*$$

edges. Thus, I^* is a yes-instance of MSE. \square

7 W[1]-hardness with Respect to Treewidth

In this section, we present the following result.

Theorem 15. MINIMUM SHARED EDGES is W[1]-hard when parameterized by treewidth and the number k of shared edges combined.

To prove Theorem 15, we give a parameterized reduction from the following problem. Herein, $\dot{\cup}$ denotes the disjoint union of sets.

MULTICOLORED CLIQUE (MCC)

Input: An undirected, k -partite graph $G = (V = V_1 \dot{\cup} \dots \dot{\cup} V_k, E)$ with $k \in \mathbb{N}$.

Question: Is there a set $C \subseteq V$ of vertices such that $G[C]$ is a k -clique in G ?

MCC is W[1]-complete when parameterized by k [11]. In the remainder of the section (G, k) is an arbitrary but fixed instance of MCC. We denote $|V_i| =: n_i$ and $V_i =: \{v_1^i, \dots, v_{n_i}^i\}$ for all $i \in [k]$. We also say that G has the *color classes* $1, \dots, k$, where each color class i is represented by the vertices in V_i . We write $E_{i,j} := \{\{v, w\} \in E \mid v \in V_i, w \in V_j\}$ for the edges connecting vertices in V_i and V_j , $i, j \in [k]$.

The reduction is based on the following idea. The routes we are to allocate will be split evenly into contingents of routes for each color class by a simple gadget. For each of the color classes, we introduce a selection gadget, that contains vertices (*outputs*) that correspond to the vertices in the MCC instance. Each selection gadget will route almost all the routes in its contingent to exactly one of its outputs. The outputs will then disperse $(k - 1)$ -times a number of routes corresponding to the ID of the vertex that this output represents. In this way, the selection gadgets represent a choice of vertices, one for each color class. In order to verify that the choice represents a clique, we introduce validation gadgets, corresponding to the pairs of color classes. They will receive the routes from the outputs of the selection gadgets, that is, the “input” of the validation gadgets is a sum of two IDs. They induce a small number of shared edges only if the vertices according to the number of routes are connected. In order to achieve this, we ensure that the sum of two IDs uniquely identifies the vertices. We achieve this by using Sidon sets.

Vertex IDs based on Sidon sets. A *Sidon set* is a set $S \subseteq \mathbb{N}$ that fulfills that for each $i, j, k, \ell \in S$ holds that if $i + k = j + \ell$ then $\{i, k\} = \{j, \ell\}$. That is, the sum of any two distinct elements in S is unique. A Sidon set S with $\max_{i \in S} i \in O(|S|^3)$ can be constructed on $O(|S|)$ time [9, page 42]. As mentioned, we use a Sidon set to distinguish numbers of routes corresponding to vertices. For this purpose, we fix a Sidon set S with $|S| = |V|$ and assign to each vertex $v \in V$ an *ID* $g(v) \in S$ where g is a bijection. For technical reasons, we need the following additional properties of g (and S):

- (i) $g(v) \geq n^3$ for all $v \in V$,
- (ii) $|g(v) - g(w)| \geq n^3$ for all $v, w \in V$, $v \neq w$, and
- (iii) $|g(v) + g(w) - (g(x) + g(y))| \geq n^3$ for all $v, w, x, y \in V$, $v \neq w$, $y \notin \{v, w, x\}$.

Clearly, by adding one to each integer in the Sidon set S and then multiplying each integer by n^3 we obtain a Sidon set and a mapping g that fulfill all of the above properties simultaneously.

To enforce that only adjacent vertices are chosen in the selection gadgets, a part in a validation gadget that represents an edge must have the property that, if many routes are routed through it, then the number of routes corresponds to precisely the sum of IDs of the endpoints of the edge that is represented by this part. To do this, we have to enforce both upper and lower bounds on the sum of IDs. Upper bounds will be enforced by long parallel paths; for lower bounds, we use

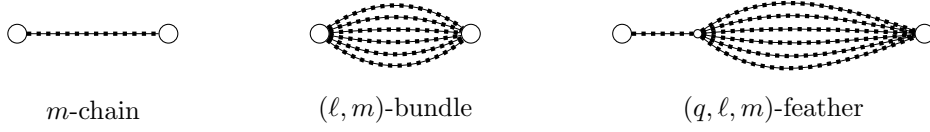


Figure 12: Illustration of a chain, bundle and feather.

the notion of “complement” of an ID. For this, we define $\overline{g(v)} := M - g(v)$ for all $v \in V$, where $M := n^3 + \max_{v \in V} g(v)$. Note that $g(v) + g(w) < g(x) + g(y)$ if and only if $\overline{g(v)} + \overline{g(w)} > \overline{g(x)} + \overline{g(y)}$ for $v, w, x, y \in V$.

Construction. In the following, we describe the construction of the instance (G', s, t, p, k') of MSE, given instance (G, k) of MCC. Initially, G' consists only of the two vertices s and t , the source and the sink vertex, respectively. We describe the gadgets we use and their interconnections, which will fully describe the construction of G' . As mentioned, our gadgetry consists of two gadget types, selection gadgets on the one hand and validation gadgets on the other hand.

Before we proceed, we fix the following notation. An m -chain is a P_{m+1} , i.e. a path of length m . A set of ℓ m -chains with common endpoints we call an (ℓ, m) -bundle. An (q, ℓ, m) -feather is obtained by identifying one endpoint of an (ℓ, m) -bundle with one endpoint of a q -chain. We also call the q -chain of the feather the q -shaft and we call the m -chains the *barbs* of the feather. Refer to Figure 12 for an illustration. In the following, by attaching a chain, bundle, or feather H to a vertex v , we mean to identify v with an endpoint of H .

We set the number of paths

$$p = \left(|E| - \binom{k}{2} \right) + k \cdot ((k-1) \cdot M + 1) + n$$

and the number of shared edges (in the following also denoted by *the budget*)

$$k' = k \cdot k^{10} + k \cdot (k + 2(k-1)) \cdot k^5 + \binom{k}{2} \cdot 3k.$$

Selection gadgets. For each color class $i \in [k]$ in the instance (G, k) , we construct a selection gadget \boxed{i} that selects exactly one vertex of V_i as follows. In Figure 13, we illustrate an example of a selection gadget \boxed{i} for color class i . We introduce vertex c_i corresponding to color class i in (G, k) . We connect s with c_i via a $((k-1) \cdot M + n_i + 1, k' + 1)$ -bundle. Each of the chains in the bundle will be in exactly one route later. We introduce the vertices $x_1^i, \dots, x_{n_i}^i$ in G' , corresponding to the vertices $v_1^i, \dots, v_{n_i}^i \in V_i$, and we connect c_i to each of them by a k^{10} -chain. These vertices serve as hubs for the routes later; only one of them will carry almost all routes in any solution, representing the choice of a vertex into the clique.

In order to relay this choice to all the validation gadgets, we do the following. First, we attach a k -chain to each vertex x_j^i , $1 \leq j \leq n_i$. Let $x_{j,1}^i, \dots, x_{j,k}^i$ denote the vertices on the chain attached to x_j^i , indexed by the distance to vertex x_j^i ; each vertex except $x_{j,k}^i$ will make its own connection to the validation gadgets. We connect each $x_{j,\ell}^i$, $\ell \in [k-1]$, with the vertex $c_i c_{\ell'}$ in the validation gadget $\boxed{i, \ell'}$ (introduced below), where $\ell' = \ell$ if $\ell < i$ and $\ell' = \ell + 1$ otherwise. The connection is made by attaching a $(k^5, g(v_j^i), k' + 1)$ -feather to $x_{j,\ell}^i$ and $c_i c_{\ell'}$. Furthermore, to relay also the complement IDs, we connect each $x_{j,\ell}^i$, $\ell \in [k-1]$, with the vertex $\overline{c_i c_{\ell'}}$ in the validation gadget $\boxed{i, \ell'}$ by attaching a $(k^5, \overline{g(v_j^i)}, k' + 1)$ -feather to them. We k^5 -subdivide each edge on the k -chain we attached to x_j^i , that is, we replace each edge by a k^5 -chain. We apply this to all paths attached to $x_1^i, \dots, x_{n_i}^i$. This will ensure that in each color class, only the “ID relay vertices” $x_{j,\ell}^i$ corresponding to one ID will carry more than one route. Note that the only differences between

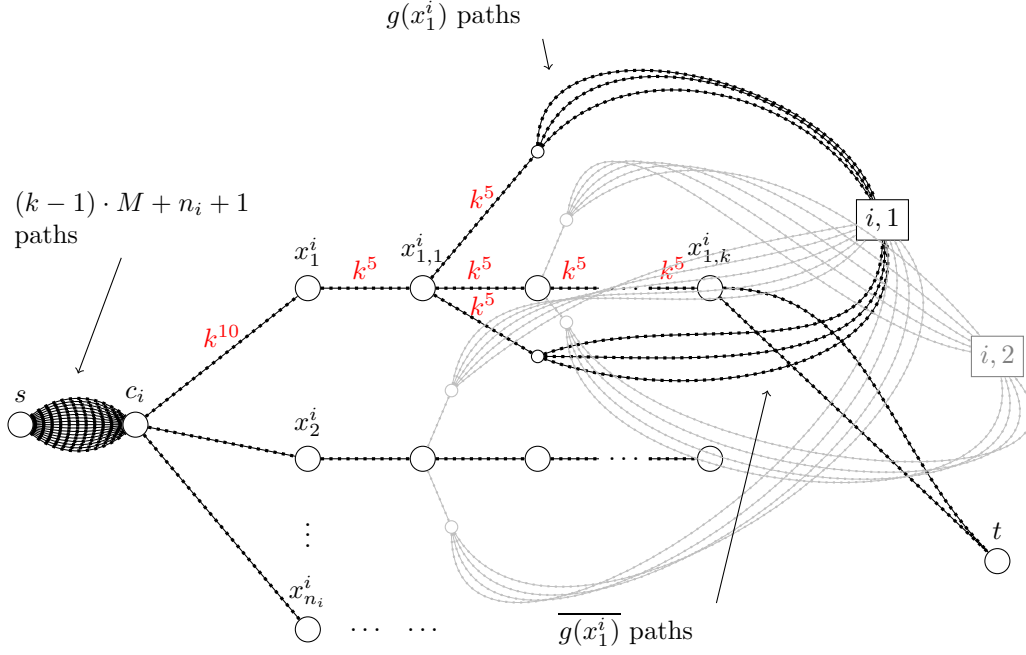


Figure 13: Example for a selection gadget $[i]$ ($i > 2$).

the ID relay vertices are the second entries of the feathers, which depend on the corresponding values of the Sidon set. Finally, we connect vertex $x_{j,k}^i$ with t via a $(2, k' + 1)$ -bundle; this vertex ensures that each k^5 -chain between two vertices $x_{j,\ell}^i$ corresponding to the chosen ID is shared.

Validation gadgets. We need to check that the chosen vertices are adjacent using only their IDs. For this we encode the sums of IDs corresponding to two adjacent vertices into a bundle which has to be passed by the routes relayed from the selection gadgets. The budget will not allow to share any of the paths in this bundle. In this way, any sum of IDs has to be below a certain threshold. To get a lower bound, we also introduce bundles for sums of complement IDs of adjacent vertices. Finally, we ensure that an “ID” bundle and its “complement ID” bundle can be used simultaneously, only if they correspond to the same pair of vertices.

We now describe the construction of a validation gadget $[i, j]$, $i, j \in [k]$, $i < j$, illustrated in Figure 14. We introduce exactly two vertices $c_i c_j$ and $\overline{c_i c_j}$ (recall that these vertices already appeared in the description of the selection gadgets). We introduce a vertex for each edge between V_i and V_j , that is, if $\{v_y^i, v_z^j\} \in E_{i,j}$, then we introduce the vertex $x_y^i x_z^j$ in G' . We connect each $x_y^i x_z^j$ to $c_i c_j$ by attaching a $(k, g(v_y^i) + g(v_z^j), k' + 1)$ -feather, we connect $x_y^i x_z^j$ to $\overline{c_i c_j}$ by attaching a $(k, \overline{g(v_y^i)} + \overline{g(v_z^j)}, k' + 1)$ -feather, and we connect $x_y^i x_z^j$ to the sink vertex t by attaching a k -chain. Only one of the connections to the sink will carry more than one route; hence, it will be possible to use only one pair of complementary bundles (corresponding to a pair of adjacent vertices).

For technical reasons, we need that each pair of bundles carries at least one route; this is achieved by also connecting s with $c_i c_j$ via an $(|E_{i,j}| - 1, k' + 1)$ -bundle.

In Figure 15, we give an overview of the interlinkage between the selection gadgets, the validations gadgets and s and t .

Correctness.

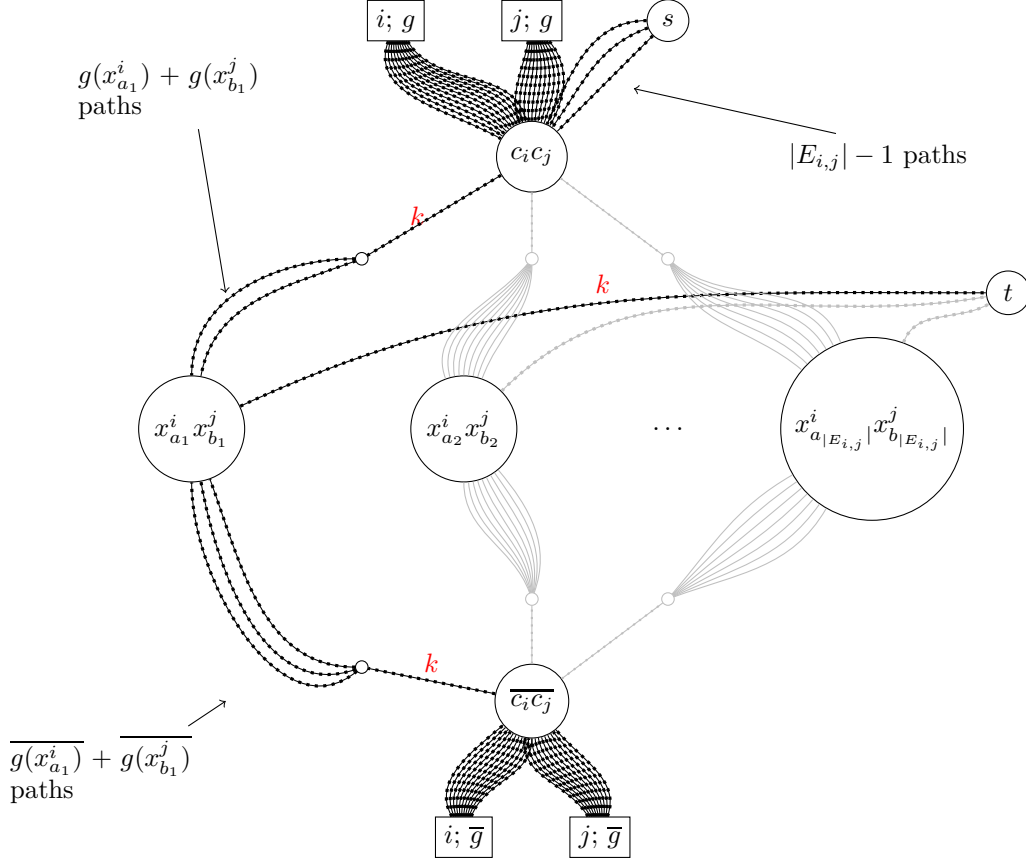


Figure 14: Example for a validation gadget $[i, j]$.

MCC yes \Rightarrow MSE yes. Suppose that (G, k) is a yes-instance of MCC, that is, G contains a k -vertex clique with vertex set W . We show that we can construct a p -routing in G' such that at most k' edges are shared.

We construct the routes in parallel, where each one starts from s , traverses the selection gadgets and then the validation gadgets. In each step, we only give a partial description of the routes up to this point. The description is then successively completed.

First, we route one route over each $(k' + 1)$ -chain incident with s . In this way, each c_i is incident with exactly $(k - 1) \cdot M + n_i + 1$ routes, and each $c_i c_j$ is incident with exactly $|E_{i,j}| - 1$ routes.

Next, we describe the routes within each selection gadget $[i]$. From c_i we route one route to each $x_{\ell'}^i$. All the remaining $(k - 1) \cdot M + 1$ routes incident with c_i are routed to the vertex $x_{\ell'}^i$ corresponding to the vertex in $V_i \cap W$. So far, this induces $k \cdot k^{10}$ shared edges overall. The routes incident with $x_{\ell'}^i$ continue as follows. Note that the barbs in the feathers incident with the $x_{\ell',j}^i$, $j \in [k - 1]$, and the bundle incident with $x_{\ell',k}^i$ count exactly $2 + \sum_{j=1}^{k-1} g(v_{\ell'}^i) + \overline{g(v_{\ell'}^i)} = 2 + (k - 1) \cdot M$ chains. This is also the number of routes incident with $x_{\ell'}^i$. We route one route incident with $x_{\ell'}^i$ over each of these chains. This induces $(k + 2(k - 1)) \cdot k^5$ further shared edges in each selection gadget. For each $x_{\ell'}^i$, $\ell \neq \ell'$, we route the single route incident with this vertex to t via $x_{\ell,k}^i$; no edge is shared in this way. Overall, we have used $k \cdot k^{10} + k \cdot (k + 2(k - 1)) \cdot k^5$ shared edges so far.

Finally, we describe the routes within each validation gadget $[i, j]$. Note that, in the way we have defined the routes so far, each $c_i c_j$ is incident with $|E_{i,j}| - 1 + g(w_i) + g(w_j)$ routes, where w_i is the vertex in $W \cap V_i$ and w_j the vertex in $W \cap V_j$. Since w_i and w_j are adjacent, there is a $(k, g(w_i) + g(w_j), k' + 1)$ -feather connecting $c_i c_j$ with some vertex $x_y^i x_z^j$ that represents the edge $\{w_i, w_j\}$. We route $g(w_i) + g(w_j)$ routes from $c_i c_j$ to $x_y^i x_z^j$ and then to t , introducing $2k$ further shared edges. The remaining $|E_{i,j}| - 1$ routes incident with $c_i c_j$ are routed via the remaining, unused feathers and then to t , without introducing more shared edges. Similarly, $\overline{g(w_i)} + \overline{g(w_j)}$

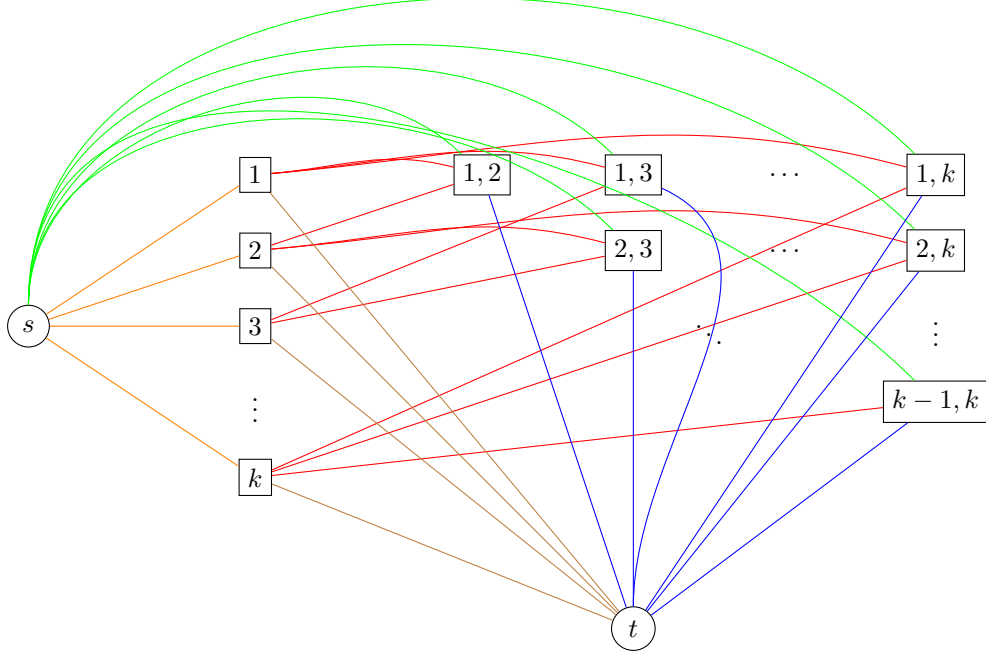


Figure 15: High-level construction.

routes are incident with $\overline{c_i c_j}$ and there is a $(k, \overline{g(w_i)} + \overline{g(w_j)}, k' + 1)$ -feather connecting $\overline{c_i c_j}$ to $x_y^i x_z^j$. We route each of the routes incident with $\overline{c_i c_j}$ over this feather to $x_y^i x_z^j$ and then to t , introducing k more shared edges (only the shaft of the feather between $\overline{c_i c_j}$ and $x_y^i x_z^j$ is additionally shared). Doing this for all validation gadgets, we introduce $\binom{k}{2} \cdot 3k$ further shared edges.

All p routes connect s and t , and exactly $k \cdot k^{10} + k \cdot (k + 2(k - 1)) \cdot k^5 + \binom{k}{2} \cdot 3k = k'$ edges are shared. Hence, we have demonstrated that a suitable p -routing exists.

MSE yes \Rightarrow MCC yes. Suppose that there is a p -routing \mathcal{P} with k' shared edges in G' . We consider the routes in \mathcal{P} as directed from s to t . Observe that s is incident with exactly p $(k' + 1)$ -chains and thus, each $(k' + 1)$ -chain is in exactly one route. This leads to the following Theorem 16.

Observation 16. *Each $(k' + 1)$ -chain incident with s is in exactly one route in \mathcal{P} . Each vertex c_i , $i \in [k]$, appears in at least $(k - 1) \cdot M + n_i + 1$ routes and each vertex $c_i c_j$, $1 \leq i < j \leq k$, appears in at least $|E_{i,j}| - 1$ routes.*

Each c_i is connected to n_i vertices corresponding to the vertices in V_i via k^{10} -chains. By Theorem 16, at each c_i , $(k - 1) \cdot M + n_i + 1$ routes in \mathcal{P} are distributed over n_i vertices. Thus, at least one of the k^{10} -chains is shared by at least two routes in for each selection gadget \boxed{i} . Since our budget k' does not allow for $(k + 1) \cdot k^{10}$ shared edges, also exactly one k^{10} -chain is shared in each selection gadget. Hence, in each selection gadget, there is exactly one vertex $x_{\ell_i}^i$ that is incident with at least two routes in \mathcal{P} . Denote by W the set of vertices in G that these $x_{\ell_i}^i$ correspond to. Clearly, W is of size k . We claim furthermore that W is a clique.

To show the claim, we use the following.

Observation 17. *For each $i \in [k]$, vertex $x_{\ell_i}^i$ is incident with exactly $(k - 1) \cdot M + 2$ routes in \mathcal{P} . Moreover, each of these routes traverses first $x_{\ell_i}^i$ and then exactly one barb in a feather incident with any $x_{\ell_{i,j}}^i$, $j \in [k - 1]$.*

To see this, denote by \mathcal{C}_i the set containing the barbs in the feathers incident with the $x_{\ell_i, j}^i$, $j \in [k-1]$, and the chains in the bundle incident with $x_{\ell_i, k}^i$. Note that

$$|\mathcal{C}_i| = 2 + \sum_{j=1}^{k-1} g(v_{\ell_i}^i) + \overline{g(v_{\ell_i}^i)} = 2 + (k-1) \cdot M.$$

Any route that contains $x_{\ell_i}^i$ either first traverses $x_{\ell_i}^i$ and then a chain in \mathcal{C}_i or vice versa, because the selection gadgets are trees except for the feathers they contain. Since each of these chains in \mathcal{C}_i has length $k'+1$, no chain can carry two routes. Thus, $x_{\ell_i}^i$ is incident with at most $2 + (k-1) \cdot M$ routes. Since each selection gadget gets at least $(k-1) \cdot M + n_i + 1$ routes via c_i , of which at most $n_i - 1$ can avoid $x_{\ell_i}^i$, each $x_{\ell_i}^i$ gets also exactly $2 + (k-1) \cdot M$ routes. Thus, indeed Theorem 17 holds.

Theorem 17 implies that each shaft of a feather incident with some $x_{\ell_i, j}^i$, $j \in [k-1]$ is shared, and each k^5 -chain connecting two $x_{\ell_i, j}^i$, $j \in [k-1]$, is also shared. The shared edges within the selection gadgets thus amount to at least $k \cdot k^{10} + k \cdot (k+2(k-1)) \cdot k^5$, leaving a budget of at most $\binom{k}{2} \cdot 3k$.

Let w_i be the vertex in W corresponding to $x_{\ell_i}^i$, $i \in [k]$ (remember that $x_{\ell_i}^i$ is the vertex in selection gadget \boxed{i} that carries at least two routes of \mathcal{P}). By Theorem 17 each $c_i c_j$ appears in at least $g(w_i) + g(w_j)$ routes in \mathcal{P} and by Theorem 16, $c_i c_j$ appears in $|E_{i,j}| - 1$ further routes in \mathcal{P} . We claim that out of these $g(w_i) + g(w_j) + |E_{i,j}| - 1$ routes, at most $n_i + n_j - 2$ routes traverse $c_i c_j$ and then, later on, traverse some vertex in a selection gadget. Call such routes *unbehaved*. To see the claim, observe that each unbehaved route has to traverse a feather incident with some $x_{\ell_i, j}^i$ or $x_{\ell_j, i}^j$. However, it cannot traverse the feathers incident with $x_{\ell_i, j}^i$ or $x_{\ell_j, i}^j$ because that would mean that one of their barbs were shared by Theorem 17. Furthermore, at most $n_i + n_j - 2$ unbehaved routes can traverse a feather incident with some $x_{\ell_{\ell}, o}^i$, $\ell \neq \ell_i$, as otherwise a shaft of one of these feathers would be shared. This contradicts our remaining budget of $\binom{k}{2} \cdot 3k$. Thus, each $c_i c_j$ has at least $g(w_i) + g(w_j) + |E_{i,j}| - 1 - (n_i + n_j - 2)$ behaved routes. Denote their number by $r_{i,j}$ and observe that each behaved route containing $c_i c_j$ traverses only vertices of the validation gadget $\boxed{i, j}$.

By the same arguments as above, the number $\overline{r_{i,j}}$ of routes that contain $\overline{c_i c_j}$ and then traverse only vertices of the validation gadget $\boxed{i, j}$ is at least $\overline{g(w_i)} + \overline{g(w_j)} - (n_i + n_j - 2)$. Note that, hence, at least $r_{i,j} + \overline{r_{i,j}}$ routes go from $c_i c_j$ and $\overline{c_i c_j}$ to t within the validation gadget $\boxed{i, j}$.

By the properties of the mapping g , it holds that $r_{i,j}, \overline{r_{i,j}}, r_{i,j} + \overline{r_{i,j}} > n^2$. Since there are strictly less than n^2 edges in $E_{i,j}$, at least two k -shafts and at least one k -chain is thus shared in the validation gadget $\boxed{i, j}$. Together with the remaining budget of $\binom{k}{2} \cdot 3k$, this observation implies that there are exactly two k -shafts and exactly one k -chain shared in the validation gadget $\boxed{i, j}$. Hence, there is exactly one vertex corresponding to an edge in $E_{i,j}$ that appears in at least two routes. Denote this vertex by $x_{y_i}^i x_{z_j}^j$.

Denote by $r'_{i,j}$ the number of routes that traverse $c_i c_j$ and then directly use the feather incident with $x_{y_i}^i x_{z_j}^j$ and denote by $\overline{r'_{i,j}}$ the number of routes that traverse $\overline{c_i c_j}$ and then directly use the feather incident with $x_{y_i}^i x_{z_j}^j$. Since $x_{y_i}^i x_{z_j}^j$ is incident with the only feathers in validation gadget $\boxed{i, j}$ whose shafts are shared, $r'_{i,j} \geq r_{i,j} - |E_{i,j}| + 1$. Similarly, $\overline{r'_{i,j}} \geq \overline{r_{i,j}} - |E_{i,j}| + 1$.

Since none of the barbs in the validation gadgets are shared, we have $r'_{i,j} \leq g(w'_i) + g(w'_j)$ and $\overline{r'_{i,j}} \leq \overline{g(w'_i)} + \overline{g(w'_j)}$ for two adjacent vertices $w'_i \in V_i$ and $w'_j \in V_j$. That is,

$$\begin{aligned} g(w_i) + g(w_j) - (n_i + n_j - 2) &\leq r'_{i,j} \leq g(w'_i) + g(w'_j), \text{ and} \\ \overline{g(w_i)} + \overline{g(w_j)} - ((n_i + n_j - 2) + |E_{i,j}| - 1) &\leq \overline{r'_{i,j}} \leq \overline{g(w'_i)} + \overline{g(w'_j)}, \text{ implying that} \\ g(w'_i) + g(w'_j) - ((n_i + n_j - 2) + |E_{i,j}| - 1) &\leq g(w_i) + g(w_j). \end{aligned}$$

We have $((n_i + n_j - 2) + |E_{i,j}| - 1) \leq n^2 + 2n < n^3$ for each $n > 2$ and hence,

$$|(g(w'_i) + g(w'_j)) - (g(w_i) + g(w_j))| < n^3.$$

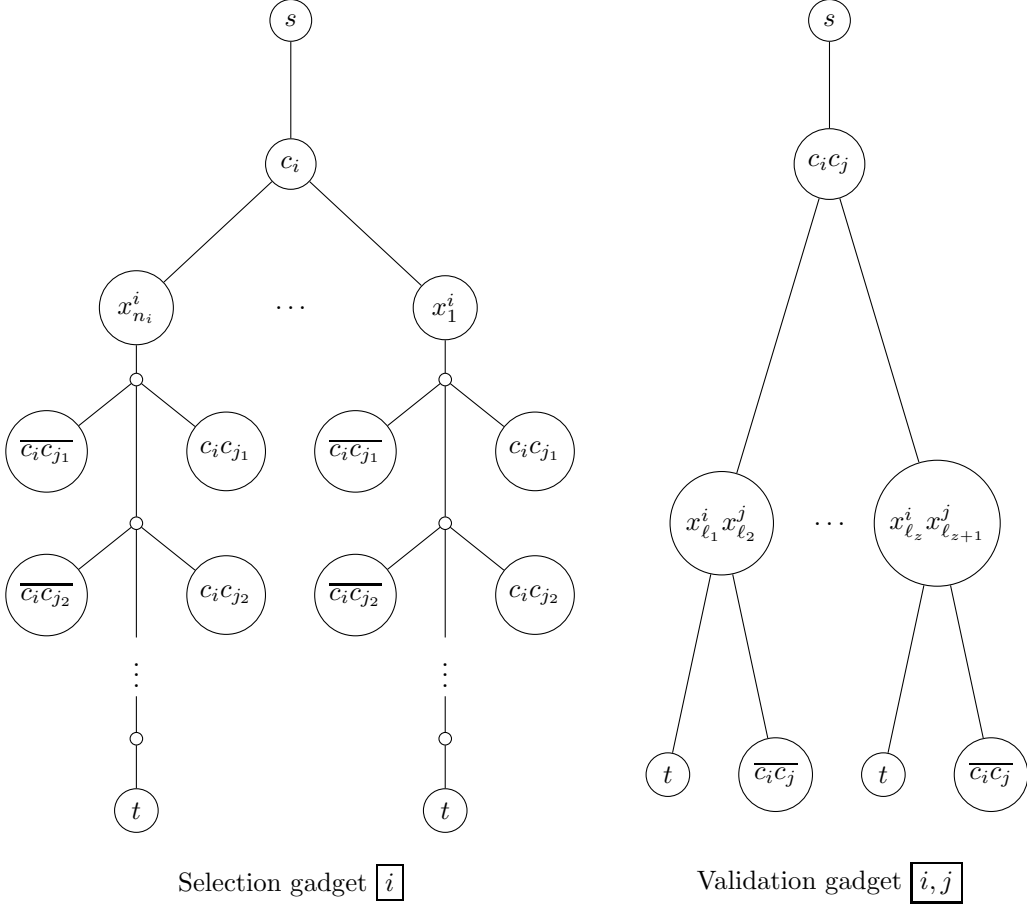


Figure 16: Tree-like structures of the selection gadget \boxed{i} and the validation gadget $\boxed{i, j}$.

Thus, by Property (iii) of g we have $\{w_i, w_j\} = \{w'_i, w'_j\}$ implying that $\{w_i, w_j\} \in E_{i,j}$. It follows that there is an edge in G between any pair of vertices in W , and W contains k vertices of each color class. Thus, $G[W]$ is a k -vertex clique.

Upper-Bound on the Treewidth. To construct a tree decomposition of small width, we start out with a single bag A , where $A := \{s\} \cup \{t\} \cup \{c_i \mid i \in [k]\} \cup \{c_i c_j \mid 1 \leq i < j \leq k\} \cup \{\overline{c_i c_j} \mid 1 \leq i < j \leq k\}$. Note that $|A| = 2 + k + 2\binom{k}{2}$. Since all gadgets are interconnected via only vertices from the set A , in order to construct a tree decomposition for G' , we can build a tree decomposition \mathbb{T}' of each gadget separately, then add A to each of its bags, and then attach \mathbb{T}' to the bag A we started with. Observe that each chain, bundle, and feather is a series-parallel graph. Since each gadget allows a tree-like structure (cf. Figure 16) where each edge corresponds to a series-parallel graph and each leaf is contained in A , we can find a tree decomposition of width at most 4 for each gadget. Hence, the treewidth of the graph G' as constructed above is upper-bounded by $2\binom{k}{2} + k + 2 + 4$.

8 Conclusion

MINIMUM SHARED EDGES (MSE) is a fundamental NP-hard network routing problem. We focused on exact solutions for the case of undirected, general graphs and provided several classification results concerning the parameterized complexity of MSE.

It is fair to say that our fixed-parameter tractability results (based on tree decompositions and the treewidth reduction technique [21]) are still far from practical relevance. Our studies indicated, however, that MSE is a natural candidate for performing a wider multivariate complexity

analysis [12, 23] as well as studying restrictions to special graph classes. For instance, there is a simple search tree algorithm solving MSE in $O((p-1)^k \cdot (m+n)^2)$ time which might be useful in some applications [14]. Moreover, it can be shown that on unbounded undirected grids (without holes), due to combinatorial arguments, MSE can be decided in constant time after reading the input [14]. In contrast, MSE remains NP-hard when restricted to planar graphs of maximum degree four (which might be of particular relevance when studying street networks), and to directed planar graphs of maximum out- and indegree three [15]. We consider it as interesting whether the running times of known FPT-algorithms (Section 4, [1, 28]) for MINIMUM SHARED EDGES can be improved for the problem restricted to planar graphs.

In the known (pseudo) polynomial-time algorithms for graphs of bounded treewidth the exponents in the running time depend exponentially on the treewidth [1, 28]. It would be interesting to know whether a polynomial dependence is achievable.

A further line of future work is to study closely related problems and natural variants of MSE. For instance, can the positive results be transferred to the more general MINIMUM VULNERABILITY problem [2] (see the introductory section)? There are also some preliminary investigations concerning the problem SHORT MINIMUM SHARED EDGES (with an additional upper bound on the maximum length of a route) [14]. Finally, it is natural to study “time-sharing” aspects for the shared edges, yielding a further natural variant of MSE.

Recently, Gutin et al. [18] proved that the Mixed Chinese Postman Problem is W[1]-hard with respect to the treewidth of input graph, but fixed-parameter tractable with respect to the tree-depth of the input graph. Since we showed that MSE is W[1]-hard with respect to the treewidth of the input graph, we consider tree-depth as an interesting parameter for a parameterized complexity analysis.

References

- [1] Yusuke Aoki, Bjarni V. Halldórsson, Magnús M. Halldórsson, Takehiro Ito, Christian Konrad, and Xiao Zhou. The minimum vulnerability problem on graphs. In *Proc. 8th International Conference on Combinatorial Optimization and Applications (COCO ’14)*, volume 8881 of *LNCS*, pages 299–313. Springer, 2014. 2, 33
- [2] Sepehr Assadi, Ehsan Emamjomeh-Zadeh, Ashkan Norouzi-Fard, Sadra Yazdanbod, and Hamid Zarrabi-Zadeh. The minimum vulnerability problem. In *Proc. 23rd International Symposium on Algorithms and Computation (ISAAC ’12)*, volume 7676 of *LNCS*, pages 382–391. Springer, 2012. 2, 33
- [3] René van Bevern, Andreas Emil Feldmann, Manuel Sorge, and Ondřej Suchý. On the parameterized complexity of computing balanced partitions in graphs. *Theory of Computing Systems*, 57(1):1–35, 2015. 2
- [4] Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Kernelization lower bounds by cross-composition. *SIAM Journal on Discrete Mathematics*, 28(1):277–305, 2014. 24
- [5] Rajesh Chitnis, Lszl Egri, and Dniel Marx. List H -coloring a graph by removing few vertices. In *Proc. 21st European Symposium on Algorithms (ESA ’13)*, volume 8125 of *LNCS*, pages 313–324. Springer, 2013. 2
- [6] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. 3
- [7] Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *Proc. IEEE 52nd Annual Symposium on Foundations of Computer Science, (FOCS 2011)*, pages 150–159, 2011. 3
- [8] Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, 4th edition, 2010. 3

- [9] Apostolos Dimitromanolakis. Analysis of the Golomb ruler and the Sidon set problems, and determination of large, near-optimal Golomb rulers. Master’s thesis, Department of Electronic and Computer Engineering, Technical University of Crete, June 2002.
<http://www.cs.toronto.edu/~apostol/golomb/>. 26
- [10] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. 3, 5
- [11] Michael R. Fellows, Danny Hermelin, Frances A. Rosamond, and Stéphane Vialette. On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science*, 410(1):53–61, 2009. 26
- [12] Michael R. Fellows, Bart M. P. Jansen, and Frances A. Rosamond. Towards fully multivariate algorithmics: Parameter ecology and the deconstruction of computational complexity. *European Journal of Combinatorics*, 34(3):541–566, 2013. 33
- [13] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Springer, 2006. 3
- [14] Till Fluschnik. The parameterized complexity of finding paths with shared edges. Master thesis, Institut für Softwaretechnik und Theoretische Informatik, TU Berlin, 2015.
<http://fpt.akt.tu-berlin.de/publications/theses/MA-till-fluschnik.pdf>. 1, 33
- [15] Till Fluschnik and Manuel Sorge. The minimum shared edges problem on planar graphs. *CoRR*, abs/1602.01385, 2016. <http://arxiv.org/abs/1602.01385>. 2, 33
- [16] Andrew V. Goldberg and Satish Rao. Flows in undirected unit capacity networks. *SIAM Journal on Discrete Mathematics*, 12(1):1–5, 1999. 4
- [17] Gregory Gutin, Mark Jones, and Bin Sheng. Parameterized complexity of the k -arc chinese postman problem. In *Proc. 22nd European Symposium on Algorithms (ESA ’14)*, volume 8737 of *LNCS*, pages 530–541. Springer, 2014. 2
- [18] Gregory Gutin, Mark Jones, and Magnus Wahlström. Structural parameterizations of the mixed chinese postman problem. In *Proc. 23rd Annual European Symposium on Algorithms (ESA ’15)*, pages 668–679, 2015. 33
- [19] Jon M. Kleinberg and Éva Tardos. *Algorithm Design*. Addison-Wesley, 2006. 4
- [20] Ton Kloks. *Treewidth, Computations and Approximations*, volume 842 of *Lecture Notes in Computer Science*. Springer, 1994. 3
- [21] Dániel Marx, Barry O’Sullivan, and Igor Razgon. Finding small separators in linear time via treewidth reduction. *ACM Transactions on Algorithms*, 9(4):30, 2013. 2, 17, 18, 21, 24, 32
- [22] Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006. 3
- [23] Rolf Niedermeier. Reflections on multivariate algorithmics and problem parameterization. In *Proc. 27th International Symposium on Theoretical Aspects of Computer Science (STACS ’10)*, volume 5 of *LIPICs*, pages 17–32. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2010. 33
- [24] Masoud T. Omran, Jörg-Rüdiger Sack, and Hamid Zarrabi-Zadeh. Finding paths with minimum shared edges. *Journal of Combinatorial Optimization*, 26(4):709–722, 2013. 1, 2, 5
- [25] Andrzej Pelc. Fault-tolerant broadcasting and gossiping in communication networks. *Networks*, 28(3):143–156, 1996. 2
- [26] Neil Robertson and Paul D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3):309–322, 1986. 21, 24
- [27] Douglas B. West. *Introduction to Graph Theory*. Prentice Hall, 2 edition, 2000. 3

- [28] Zhi-Qian Ye, Yi-Ming Li, Hui-Qiang Lu, and Xiao Zhou. Finding paths with minimum shared edges in graphs with bounded treewidths. In *Proc. Frontiers of Computer Science (FCS '13)*, pages 40–46, 2013. 2, 33